

EVELIT



ESD-TR-81-118

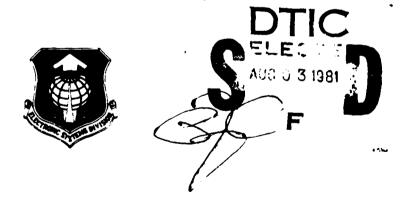
MICROCOMPUTER POLLING IMPROVEMENTS FOR AFSATCOM

JACOB HANDWERKER

JUNE 1981

Prepared for

DEPUTY FOR COMMUNICATIONS AND INFORMATION SYSTEMS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts



C FILE COPY

Approved for public release; distribution unlimited,

Project No. 6340
Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-81-C-0001

81 7 31 074

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

(Signature)

Joseph Mardo/GS-13

Project Engineer

FOR THE COMMANDER

(Signature)

MAX I. MILLER, Jr., Colonel, USAF

System Program Director

Satellite Communications Terminal SPO

Deputy for Communications and

Information Systems

	READ INSTRUCTIONS BEFORE COMPLETING FORM
	NO. 3. RECIPIENT'S CATALOG NUMBER
ESD TR-81-118 - A7-HZ	02 394
4. TITLE (and Submitte)	TYPE OF REPORT & PERIOD COVER
MICROCOMPUTER POLLING IMPROVEMENTS	W. G. Commerce Contraction
FOR AFSATCOM	The same of the sa
Company of the Compan	PERFORMING DAG, REPORT NUMBER
	MTR-8239 (
7. AUTHOR(a)	S. CONTRACT OR GRANT NUMBER(a)
Jacob/Handwerker /	F19628-81-C-0001 4
and the second second	7
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TAS
The MITRE Corporation	AREA & WORK UNIT NUMBERS
P.O. Box 208	Project No. 6340
Bedford, MA 01730	
11. CONTROLLING OFFICE NAME AND ADDRESS	12: REPORT DATE
Deputy for Communications and Information Systems	/ June 1981
Electronic Systems Division, AFSC	13 NUMBER OF PAGES
Hanscom AFB, MA 01731	122 e) 15. SECURITY CLASS. (of this report)
14. MONITORING AGENCY NAME & ADDRESS(it different Acon Controlling Office	
	UNCLASSIFIED
	15a DECLASSIFICATION DOWNGRADING
The second secon	SCHEDULE
17. DISTRIBUTION STATEMENT (of the abatract entered in Block 20, if different	from Report)
17. DISTRIBUTION STATEMENT (of the ebetract entered in Block 20, if different	from Report)
17. DISTRIBUTION STATEMENT (of the abatract entered in Block 20, if different	from Report)
17. DISTRIBUTION STATEMENT (of the abatract entered in Block 20, if different	from Report)
17. DISTRIBUTION STATEMENT (of the ebetract entered in Block 20, if different	from Report)
•	from Report)
•	from Report)
•	from Report)
18. SUPPLEMENTARY NOTES	
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num	ber)
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num AFSATCOM MICROCOMPUTER POL:	ber)
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num AFSATCOM MICROCOMPUTER POL.: AFSATCOM POLLING	ber)
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num AFSATCOM MICROCOMPUTER POL: AFSATCOM POLLING MICROCOMPUTER	ber)
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num AFSATCOM MICROCOMPUTER POL.: AFSATCOM POLLING	ber)
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num AFSATCOM MICROCOMPUTER POL. AFSATCOM POLLING MICROCOMPUTER MICROCOMPUTER MICROPROCESSOR 20. ABSTRACT (Continue on reverse side if necessary and identify by block numb	ber) LING
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num AFSATCOM MICROCOMPUTER POL. AFSATCOM POLLING MICROCOMPUTER MICROCOMPUTER MICROPROCESSOR 20. ABSTRACT (Continue on reverse side if necessary and identify by block numb) This report documents a cost effective approach for a	LING ver) n on-line, remotable demonstra
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num AFSATCOM MICROCOMPUTER POL. AFSATCOM POLLING MICROCOMPUTER MICROCOMPUTER MICROPROCESSOR 20. ABSTRACT (Continue on reverse side if necessary and identify by block numb of the policy of the pol	LING or) n on-line, remotable demonstration of the contraction of th
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num AFSATCOM MICROCOMPUTER POL. AFSATCOM POLLING MICROCOMPUTER MICROPROCESSOR 20. ABSTRACT (Continue on reverse side if necessary and identify by block numb of the policy of the pol	LING n on-line, remotable demonstrate through currently available, s of present and enhanced AFSA
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num AFSATCOM MICROCOMPUTER POL. AFSATCOM POLLING MICROCOMPUTER MICROCOMPUTER MICROPROCESSOR 20. ABSTRACT (Continue on reverse side if necessary and identify by block numb) This report documents a cost effective approach for a tion of AFSATCOM polling improvements made possible eral purpose microcomputer technology. Description COM terminal capabilities, and a comprehensive over	LING n on-line, remotable demonstrate through currently available, s of present and enhanced AFSA view of the resultant microproc
18. SUPPLEMENTARY NOTES 19. KEY WORDS (Continue on reverse side if necessary and identify by block num AFSATCOM MICROCOMPUTER POL. AFSATCOM POLLING MICROCOMPUTER MICROPROCESSOR 20. ABSTRACT (Continue on reverse side if necessary and identify by block numb of the policy of the pol	LING n on-line, remotable demonstrate through currently available, s of present and enhanced AFSA view of the resultant microproc

DD 1 JAN 73 1473

UNCLASSIFIED

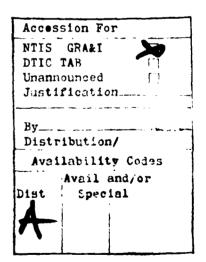
SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered

20. ABSTRACT (Continued)	
hetting capabilities for the satellite-user con	
	\prec
	<i>'</i>
	:
	j

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ABSTRACT

This report documents a cost effective approach for an on-line, remotable demonstration of AFSATCOM polling improvements made possible through currently available, general purpose microcomputer technology. Descriptions of present and enhanced AFSATCOM terminal capabilities, and a comprehensive overview of the resultant microprocessor-based hardware and software, and the support facility, are presented. The conclusion is that message processor unit software-only changes or stand-alone microcomputer hardware additions to existing AFSATCOM terminals can significantly enhance netting capabilities for the satellite-user community.



ACKNOWLEDGMENTS

Grateful acknowledgment is made to Robert K. Waite whose software and hardware expertise and general participation in the microcomputer project described herein were instrumental to its success. Also, recognition is given to Anthony J. Lauletta for his mechanical and electronic skills in assembly of the system, and to Fay S. Durk for providing needed support and understanding of MITRE's Time Sharing Option (TSO) system. Finally, acknowledgment is made to William T. Higgins for his unwavering managerial support which ensured the success of this project.

This report has been prepared by The MITRE Corporation under Project 6340. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

Section			Page
	LIST	OF ILLUSTRATIONS	viii
	LIST	OF TABLES	x
1	INTR	ODUCTION	1
2	AFSA	TCOM SYSTEM OVERVIEW	3
	2.1	SPACE SEGMENT	3
	2.2	TERMINAL SEGMENT	5
	2.3	AFSATCOM TYPE 12 COMMAND POST TERMINAL	6
	2.4	CURRENT AFSATCOM POLLING CAPABILITIES	9
	2.5	POLLING LIMITATIONS	9
3	AFSA	TCOM POLLING IMPROVEMENTS PROJECT	13
	3.1	PROJECT DESCRIPTION	13
	3.2	LIMITED TEST RESULTS	15
4	MICR	OCOMPUTER-BASED POLLING ENHANCEMENTS	16
	4.1	MAJOR POLLING IMPROVEMENTS	16
		4.1.1 Group/Non-Group Polling and Prioritization	16
		4.1.2 Remote Entry Automatic Check-in	18
		4.1.3 TDM Auto Polling	19
		4.1.4 Automatic Slot Assignment	19
		4.1.5 Encrypted Non-TDM Polling	20
		4.1.6 Error Detection and Correction (EDAC)	20
		4.1.7 Variable Non-Reportback Time-out	21

TABLE OF CONTENTS (continued)

Section			<u>Page</u>
	4.2	NEW COMMANDS AND MESSAGES FOR POLLING	21
		4.2.1 COMSUP Commands	21
		4.2.2 Error Messages	26
		4.2.3 External Message Formats	26
5	MICR	OCOMPUTER HARDWARE	28
	5.1	INTEL SBC 80/20 SINGLE BOARD COMPUTER	28
	5.2	NATIONAL SEMICONDUCTOR BLC 416 16K PROM BOARD	32
	5.3	INTEL 450 16K RAM BOARD	32
	5.4	I/O AND DISPLAY CIRCUITS	32
		5.4.1 Master Clock Generator	32
		5.4.2 External I/O Interface	35
		5.4.3 RX Character Clock Generator	35
		5.4.4 Memory I/O Address Decoder/Acknowledge	35
		5.4.5 TX Character Deletion	35
		5.4.6 Modem/MPU-to-SBC Serial Data Adapter	40
		5.4.7 SBC 80/20 Interfaces	40
		5.4.8 Slot Counter Display	40
		5.4.9 RS-232 Serial Interfaces	44
	5.5	MODULAR RACK HARDWARE FEATURES	44
6	MICR	OCOMPUTER SOFTWARE	46
	6 1	FIOW CHAPTS	46

TABLE OF CONTENTS (concluded)

Section			Page
	6.2	PL/M-80 Source Code and Memory Allocation	49
	6.3	MITRE TSO SYSTEM PL/M-80 SUPPORT	49
		6.3.1 PL/M-80 High Level Language	49
		6.3.2 PL/M-80 2-Pass Cross-Compiler	49
		6.3.3 INTERP/80 for the 8080 Microprocessor	63
	6.4	TEKTRONIX 8002A SOFTWARE DEVELOPMENT SYSTEM	63
7	CONC	LUSIONS/RECOMMENDATIONS	69
LIST OF	REFE	CRENCES	72
APPENDIX	< A	BASELINE DESCRIPTION OF THE AFSATCOM ROLL CALL	
		POLLING MODE	73
APPENDIX	КВ	PL/M-80 SOURCE CODE	81
GLOSSARY	Z.		111

LIST OF ILLUSTRATIONS

Figure		Page
2-1	AFSATCOM System Concept	4
2-2	Type 12 Ground Command Post Terminal Functional Block Diagram	7
2-3	Polling Delay Performance	12
3-1	Polling Hardware/Software Design Cycle	14
5 - 1	Microcomputer Functional Block Diagram	29
5-2	Microcomputer System Assembly	30
5 - 3	AFSATCOM T12-to-Microcomputer System Interface Diagram	33
5 - 4	Master Clock Generator (Sheet 1)	34
5-5	External I/O Interface (Sheet 2)	36
5 -6	RX Character Clock Generator (Sheet 3)	37
5-7	Memory I/O Address Decoder/Acknowledge (Sheet 4)	38
5-8	TX Character Deletion (Sheet 5)	39
5 -9	Modem/MPU-to-SBC Serial Data Adapter (Sheet 6)	41
5-10	SBC 80/20 Single Board Computer Interfaces (Sheet 7)	42
5-11	Slot Counter/Display (Sheet 8)	43
5-12	RS-232 Serial Interfaces	45
6-1	Microcomputer Development Facility	47
6-2	Polling Software Program Flowchart	48
6-3	Polling Algorithm Flowchart	54

LIST OF ILLUSTRATIONS (concluded)

<u>Figure</u>		Page
6-4	File Structure and Flow of Program Execution on 8080 PL/M Cross-Compiler	61
6-5	Run-Time Storage Organization of Memory Storage Allocation	62
6-6	Tektronix 8002A Microprocessor Laboratory Work Station	65

LIST OF TABLES

<u>Table</u>		Page
4-1	Improved Polling Performance Comparison	17
4-2	COMSUP Commands for Polling Enhancements	22
4-3	Error Message Formats	27
4-4	External Message Formats	27
5-1	EPROM (2708) Location Summary	31
6-1	Program Software Module Description	50
6-2	Symbol Table and Memory Map	55
6-3	INTERP/80 Commands	64

SECTION 1

INTRODUCTION

This report describes a breadboard microcomputer system based on an Intel SBC 80/20 Single Board Computer (SBC) which fulfills the project demonstration requirements for an improved polling capability for various satellite terminal users. The system underscores the flexibility and power of microprocessor-based nardware and software applications and provides extensive simulation capabilities related to terminal performance improvements for Air Force Satellite Communications (AFSATCOM) ultra high frequency (UHF) satellite communications in a real-time environment.

A preliminary study to investigate the attributes and growth potential of the AFSATCOM narrowband (NB) automatic Roll-Call Polling mode was undertaken by The MITRE Corporation for the Electronic Systems Division/Air Force Systems Command (ESD/AFSC). The purpose of this study was to lay the groundwork for eventual implementation of polling mode enhancements through software-only changes to the AFSATCOM message processor unit (MPU) subsystem as one means of improving overall netting capabilities. Subsequent polling-related studies involving advanced concepts have identified growth possibilities in the MPU subsystem which need software-only changes.

As a result of these studies, it was concluded that many of these new polling capabilities could be demonstrated easily and economically through use of currently available microcomputer techniques without the need to modify any existing AFSATCOM hardware or software. New functional capabilities could be added by a serially transparent microcomputer using existing interfaces presently employed by the AFSATCOM MPU subsystem. A polling improvements project was therefore undertaken which resulted in the development and testing of the microcomputer system detailed in this document.

さると きるいちゅうしん

To place the polling improvements in perspective, an overview of the AFSATCOM space and terminal segments is presented in section 2. Section 3 briefly describes the polling improvements project and its initial test results. Section 4 presents details of the polling improvements actually implemented by using a microcomputer add-on enhancement with an AFSATCOM terminal. Sections 5 and 6 provide details of the microcomputer hardware and software, respectively, for the demonstration system. Section 7 presents the overall conclusions and recommendations with regard to AFSATCOM polling enhancements and the follow-on objectives and planning for

subsequent demonstration testing. The AFSATCOM Roll-Call Polling mode is described in appendix A and a full source code listing is provided in appendix B.

SECTION 2

AFSATCOM SYSTEM OVERVIEW

The AFSATCOM System is designed to provide command, control, and communications capability on a worldwide basis to Single Integrated Operations Plan (SIOP) and other designated high priority users for emergency action message (EAM) dissemination, force direction, force reportback, and Commander-in-Chief (CINC) internetting. Communications are also provided for a limited number of non-SIOP normal force elements. The AFSATCOM System operates in the 225 to 400 MHz UHF spectrum and uses the 75 bits-per-second (b/s) teletype (TTY) service with frequency shift keying (FSK) modulation.

The AFSATCOM System, depicted in figure 2-1, consists of a space segment and a terminal segment. The space segment includes a communications capability designed into satellites such as the Navy Fleet Satellite Communications (FLTSATCOM) satellites as well as Air Force transponders carried "piggy-back" on other vehicles. The terminal segment is composed of various configurations operationally characterized as SIOP, normal force (non-SIOP), and command. SIOP and normal force terminals utilize only 5-kHz NB FSK channels, while command terminals also have access to an M-ary FSK 500-kHz bandwidth wideband spectrum. To ensure orderly management of day-to-day operation and control, each terminal user is part of an overall AFSATCOM System control structure.

2.1 SPACE SEGMENT

The AFSATCOM space segment consists of several types of transponder carried on different host satellites. AFSATCOM equipment is designed into each of the Navy FLTSATCOM satellites in geosynchronous equatorial orbit to provide overlapping earth coverage in all areas except the polar regions. Polar coverage is provided by Satellite Data System (SDS) satellites placed in highly inclined elliptical orbits.

Each satellite has 12 NB channels for specific user functions. The first seven NB channels are used to disseminate the EAM, force direction, and force reportback for SIOP/nuclear-capable forces. The next four NB channels are assigned to support essential operations of non-SIOP high priority Air Force users. The twelfth NB channel is used for system control and is designed to function as the orderwire (OW). Each of the FLTSATCOM satellites also has a

A SALES TO SALES

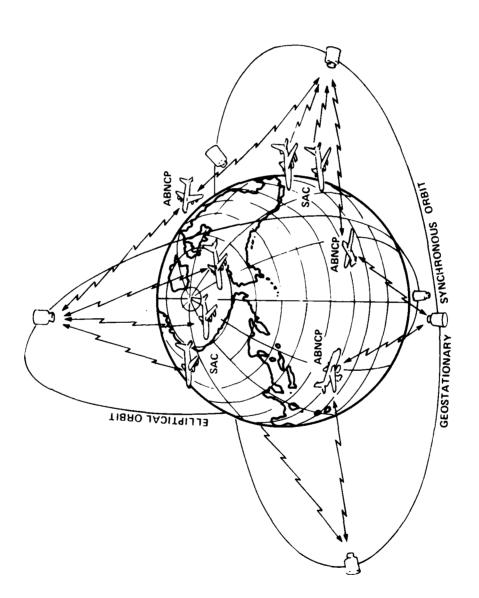


Figure 2-1 AFSATCOM System Concept

wideband (WB) channel. Accesses on the WB channel have been assigned to the National Command Authority (NCA), the CINCs, and members of the Worldwide Airborne Command Post (WWABNCP) fleet for the CINC Internet. SIOP and force terminals do not have access to the WB channel. Command post and net control terminal elements have the capability to command the various satellite modes using their OW or NB channel equipment.

2.2 TERMINAL SEGMENT

The AFSATCOM terminal segment consists of various equipment configurations to meet SIOP, normal force, and command operational requirements. The terminals are airborne or ground-based depending on the user application.

SIOP terminals provide 75 b/s half-duplex TTY operation via the 5-kHz NB satellite channels and can be operated in time division multiplex (TDM) modes under control of a command post. The input/output (I/O) devices are a keyboard, teleprinter, and appropriate control/monitor panels. Hard copy is available from terminals using the teleprinter. A SIOP synchronizer provides proper synchronization during TDM operation.

Normal force terminals (airborne and ground) provide 75 b/s half-duplex TTY operation via a 5-kHz, fixed frequency satellite channel. The absence of a SIOP synchronizer allows the normal force terminals to operate only in non-TDM modes.

Command post (CP) terminals can send, receive, and monitor messages; establish SIOP timing synchronization for TDM-1 (normal) and TDM-2 (stressed) modes; provide satellite UHF commands; initiate EAMs; and operate in a network control system using an OW channel. They can also monitor and record all reportback data to allow command transition. These terminals use full-duplex 75 b/s TTY operation on the 500-kHz WB channel, along with half-duplex and full-duplex 75 b/s TTY on the 5-kHz NB channels. Each CP terminal can operate as either a CP terminal or SIOP force terminal when in a slave mode. In the slave mode, SIOP timing synchronization is provided by another CP terminal. The AFSATCOM Type 12 CP terminal is described below since it was used for the polling improvement demonstration capability discussed later.

2.3 AFSATCOM TYPE 12 COMMAND POST TERMINAL

The AFSATCOM Type 12 (AN/TSC-88) CP terminal provides full-duplex record communications using FSK modulation at 75 b/s and is housed in an S-280 shelter that can be transported for rapid global deployment. The terminal consists of equipment installed in four electrical equipment cabinets and a two-position operator console, all within the shelter. Steerable and fixed antennas are set up outside the shelter when it is operational and are stowed inside during transport. Figure 2-2 is a functional block diagram of the Type 12 CP terminal.

Five AN/ARC-171 transceivers (R/Ts) and their associated controls, in conjunction with two half-duplex and one full-duplex narrowband modems, and two full-duplex wideband modems, provide five transmit (TX) and 12 receive (RX) communications channels. These channels are identified as follows:

- NB-1: a one transmit/one receive (1x1), half-duplex, NB (5-kHz) channel.
- NB-2: a one transmit/eight receive (1x8), full-duplex, NB (5-kHz) channel.
- OW: a one transmit/one receive (1x1), half-duplex, NB (5-kHz) channel.
- WB-1: a one transmit/one receive, full-duplex, WB (500-kHz) channel.
- WB-2: a one transmit/one receive, full-duplex, WB (500-kHz) channel.

Several narrowband operating modes are available in the AFSATCOM System. During normal operation, messages on NB channels one through seven are received by FLTSATCOM and SDS satellite packages on fixed uplink frequencies and retransmitted on fixed downlink frequencies. Methods of transmission in the normal mode include random and polled access. Random operation allows the terminal to enter the net on a random access basis using an open (not busy) select channel. Polled operation allows the Type 12 CP terminal to query each terminal in the net in sequence, with each terminal responding automatically with a precomposed message when it detects its own unique code sequence.

All of the NB channels, including the OW channel, can be used for automatic polling. In the polling mode, the Type 12 CP terminal sequentially addresses poll inquiry messages to terminals assigned



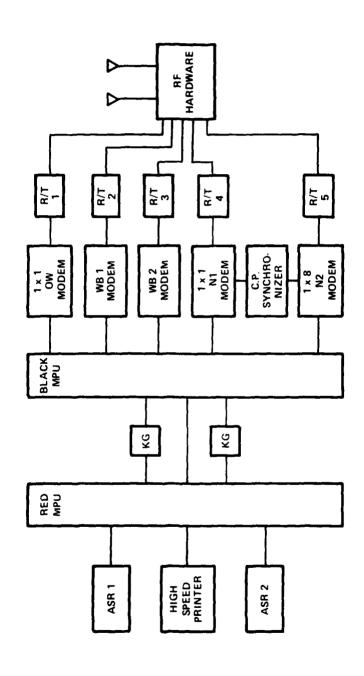


Figure 2-2 Type 12 Ground Command Post Functional Block Diagram

to various poll groups. Poll groups and polling operations are under the control of software programs in the terminal computer, the MPU. Four separate poll groups and up to 16 members apiece are allowed by the software. Each terminal to be polled will precompose a poll response message. When the unique address of that terminal is recognized in a poll message, the precomposed message of up to 30 seconds in length is automatically transmitted.

During periods of high density traffic, the Type 12 terminal can initiate synchronized operation for SIOP users by implementing a TDM mode. In this mode, the CP synchronizer periodically transmits a sync message which automatically establishes a time reference within each SIOP terminal in the net. In the TDM-1 mode, each SIOP terminal is constrained to transmit a precomposed message within its assigned time slot. Sixty consecutive time slots constitute a frame after which the sequence is repeated, with the CP synchronizer transmitting a new sync message in the last time slot of each frame.

During TDM-1, only the transmit time of each SIOP terminal is controlled by the CP synchronization message. During TDM-2, both the transmit time and transmit frequency of each SIOP terminal are synchronized by the CP synchronizer. In the TDM-2 mode, the satellite transponder is commanded into a stressed mode, which causes the first seven NB receive channels in the satellite to frequency hop on a pseudorandom basis. Also in the TDM-2 mode, each SIOP terminal is constrained to transmit in its assigned time slot and the transmit frequency changes with each time slot. The transmit frequency is determined by the code-of-the-day which is programmed into each synchronizer by the operator. Before the net can enter the TDM-2 mode, the satellite must be commanded into the stressed mode.

Three modes of operation are also provided on the WB M'ary FSK channels. These are: (1) random access precomposed message; (2) random access manual message composition; and (3) full-duplex relay, which is also available in NB random access mode.

During normal CP terminal operations, communications channels OW and WB-2 are assigned to operate with a fixed antenna. The WB-1, NB-1, and NB-2 channels are assigned to operate with a tracking antenna. Antenna switching is provided to allow the OW and WB-1 channels to operate with either antenna.

Additional details pertaining to the AFSATCOM Type 12 terminal can be found in another document.

2.4 CURRENT AFSATCOM POLLING CAPABILITIES

The present AFSATCOM polling capability is primarily structured for automatic operation in the non-TDM unencrypted mode over a single 5-kHz NB channel. Polling operations for AFSATCOM System control are conducted from MPU-equipped terminals by utilizing a polling algorithm contained in the MPU software. Communications supervisory (COMSUP) commands input by the terminal operator are used either to enter or delete poll addresses as well as to start polling operations. Up to 64 pollees may be entered into a poll net consisting of four groups having 16 members, maximum, per group. Addresses for each group member are related to others in the same group, with poll address recognition actually taking place within the AFSATCOM automatic send-receive (ASR) I/O hardware. During MPU automatic polling operations, a non-reportback of a polled address will cause a 30-second fixed time-out delay within the MPU before generation of the next poll mersage. Appendix A gives a more complete description of current AFSATCOM polling operation, including polling message formats, address structure, and poll-related COMSUP commands for the MPU.

2.5 POLLING LIMITATIONS

The following limitations of the AFSATCOM polling mode constrain its utility in an operational environment.

1. Small Network Size

A maximum of only 64 pollees can be accommodated in the existing MPU polling software. Present AFSATCOM requirements forsee nets needing to accommodate between two and three times this number.

2. Inflexible Addressing Structure

The group organization required in present auto polling dictates that all members of a poll group have the last two of their three address characters in common. Polling always commences with the lowest address of each group of 16 and proceeds in sequence toward the highest address before attempting to poll the next group. No provision is made for randomized poll addressees who are not group-related members.

3. No Prioritization/Interruption Features

No provision is made for real-time prioritization of pollees based on operational needs for early or more frequent reportback requirements. Also, no provision is made for momentary or short-term interruptions of polling operations to accommodate higher priority traffic without complete repolling of all users.

4. Inefficient Bookkeeping Scheme

Current polling operations dictate that all poll address additions or deletions be entered by COMSUP command. This can be a time consuming process, especially since no provision for a truly automated MPU poll table generation/display is available to the operator. Also, prior coordination of poll address assignments is essential but is unwieldly when attempted in real-time. Manual bookkeeping procedures are thus mandated to keep abreast of current poll net membership.

5. No Automated Handover Procedures

No provision is currently made for an automated handover procedure to an alternate net control station (NCS) for polling operations. All poll address additions/deletions must be accommodated by the alternate NCS on a manual basis.

6. No TDM Polling Mode

Automatic polling is currently restricted to the non-TDM mode. Although TDM poll response is possible from a polled I/O, the existing MPU polling software is not presently configured to cope with the structural differences between AFSATCOM non-TDM and TDM operation since automated TDM polling was not originally conceived as a basic AFSATCOM netting requirement.

7. Single Channel Polling Restriction

Polling inquiry messages sent by the MPU during polling operations are limited to a single channel. Simultaneous, independent polling on more than one channel is not currently possible in a terminal equipped with a single MPU.

8. Unencrypted Reportback Limitation

In general, the 30-second time-out constraint during automatic polling limits the length of reportback possible. In addition,

MPU-equipped terminals incur internal MPU delays such that their reportback is limited to 130 characters, maximum. This restricts the use of the Auto Polling mode to unencrypted operation with a fixed upper limit on the size of a reportback message; the upper limit depends on the type of terminal responding.

9. No-show Delay Penalty

The present fixed 30-second time-out for auto-polling also incurs a severe delay penalty when a no-show occurs. A no-show, or non-reportback, may occur for a variety of reasons such as failure to properly receive the polling inquiry message, failure to be in the I/O poll mode, etc. Figure 2-3 shows the polling delay performance of the present non-TDM roll-call polling mode as a function of the number of non-reportbacks (assuming a 40-character reportback when one does occur) for the maximum net size of 64 polled non-MPU terminals. The vertical axis indicates the time to completion for polling the entire net. As fewer net members report back, the time to completion for polling increases dramatically. This is counter to the concept of maintaining a predictable level of AFSATCOM network performance in situations supposedly under the control of a CP terminal.

10. Error Detection and Correction

では、 とうできるというないのである。 日本のはいのの

No automated procedure currently exists for repolling a net member upon detection of a character error while in the polling mode. The existing AFSATCOM unencrypted message structure provides for odd-parity character transmission, but this capability has yet to be fully exploited.

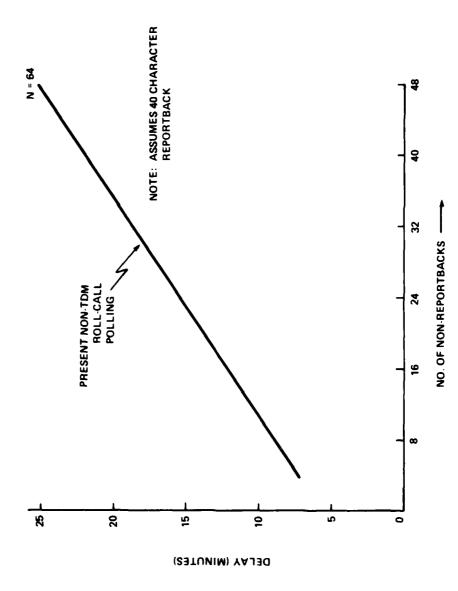


Figure 2-3 Polling Delay Performance

SECTION 3

AFSATCOM POLLING IMPROVEMENTS PROJECT

Providing enhanced polling capabilities by using AFSATCOM-compatible techniques to meet the ever increasing demands of the AFSATCOM System is a primary concern because the Air Force has an extensive hardware investment in terminal assets. Significant changes to present hardware or software must be evaluated with regard to both operational and logistical effects. Polling improvements should also provide flexibility for a wide range of applications yet to be defined and at the same time address the reportback needs of specific AFSATCOM users with known requirements.

3.1 PROJECT DESCRIPTION

A microcomputer-based approach for demonstrating various improved AFSATCOM polling capabilities, involving no hardware or software modifications to existing AFSATCOM equipment, was conceived by MITRE to be the most cost-effective means of validating many of the proposed concepts. The microcomputer provides real-time, online simulation of new polling capabilities while functionally emulating changes to MPU software. Making this microcomputer serially transparent to existing system operation with no need to modify AFSATCOM assets is an inexpensive way to validate new polling concepts. Operational MPU software does not need to be modified until after these concepts are validated and user coordination is achieved on a final implementation approach. Physically, the microcomputer interfaces between the MPU and either NB-1 or OW modem in the AFSATCOM Type 12 terminal. A project was initiated to translate these improved polling concepts into actual breadboard hardware and software suitable for demonstration purposes. For this purpose, the microcomputer development facility based around a Tektronix 8002A microprocessor laboratory at MITRE-Bedford was used.

Figure 3-1 describes the overall implementation approach for the project. Because of the small staff for this undertaking (two part-time MITRE technical staff), relatively close coordination between hardware and software design efforts was possible, with the result that the project was successfully completed in nine calendar months.

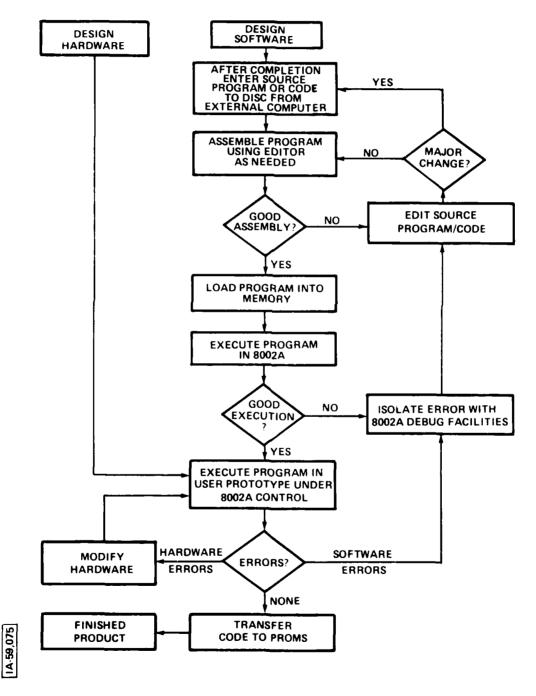


Figure 3-1 Polling Hardware/Software Design Cycle

A de lang. To all a lang.

3.2 LIMITED TEST RESULTS

and the second section of the second second

Following completion of bench testing and terminal integration, the microcomputer system was made available for demonstration testing. Technical compatibility of the improved polling hardware/ software with the AFSATCOM Type 12 terminal was demonstrated during the initial tests. With the microcomputer installed in the terminal, both software and hardware bypasses provided AFSATCOM system transparency for normal AFSATCOM modes. After completion of various terminal performance tests, actual network testing was conducted using the FLTSATCOM satellites. of the significant performance improvements for polling (described in detail in section 4) were also demonstrated successfully in conjunction with other cooperative AFSATCOM terminals operating through the satellite system. As a result, the microcomputer system is now considered available for network-wide tests on a much larger scale, with emphasis on operational testing to demonstrate improved netting capabilities to potential users.

SECTION 4

MICROCOMPUTER-BASED POLLING ENHANCEMENTS

Significant improvements to the present AFSATCOM polling mode operation have been achieved with the microcomputer-based addition to the AFSATCOM Type 12 CP terminal. These improvements have been demonstrated to participating terminal operators during actual online demonstrations.

4.1 MAJOR POLLING IMPROVEMENTS

The major polling improvements implemented in the AFSATCOM Type 12 CP terminal are described below. Table 4-1 compares existing AFSATCOM polling capabilities and the microcomputer-based improvements. These improvements are a real-time simulation of capabilities possible with software-only improvements to the AFSATCOM MPU.

4.1.1 Group/Non-Group Polling and Prioritization

The current AFSATCOM System can poll up to four groups of related net members where each group is defined as a set of 16 addresses and each address in the group has the same last two hexadecimal address characters. This group structure is retained in the improved polling software. In addition, a new set of net members is permitted which may contain up to 80 random addressees (equivalent to five 16-member groups) independent of any related group address structure. This provides flexibility to organize and poll up to a maximum of 144 group and non-group net members per channel.

During polling net organization with the microcomputer software, the non-group members are accorded a higher priority level with respect to group-related members. Four priority levels of polling net organization have been chosen for demonstration purposes: (1) priority check-ins, (2) routine check-ins, (3) notraffic check-ins, and (4) group check-ins.

During polling operations, the non-group related check-ins of priority levels 1, 2, and 3 are polled before the group-related check-ins of level 4. The NCS terminal operator can enter any of these check-ins into the polling net by COMSUP command. A remoteentry automatic check-in for levels 1, 2, and 3 is also possible.

Table 4-1 Improved Polling Performance Comparison

to do the service of the service of

The second secon

	Non-TDM	Non-TDM Polling Mode	TDM PO	TDM Polling Mode	
AFSATCOM Functional Capability	Present	With Microcomputer	Present	With Microcomputer	
Auto Polling	Yes	Yes	NO	Yes	
Encryption	No	Yes	N/A	ON O	
Limited EDAC	N _O	Yes∺	N/A	N	
Non-Group Polling/Prioritization	No	Yes	N/A	Yes	
Auto Check-in	N _O	Yes	N/A	Yes	
Automatic Slot Assignment	N/A	N/A	N/A	Yes	
Variable Time-Out	№	Yes	N/A	N/A	
Table Printout/Transfer (With Selective Routing)	NO	Yes	N/A	Yes	
Poll Interrupt/Resume	N _O	Yes	N/A	Yes	
Number of Users	**+9	144***	N/A	48/Frame	

*Poll Messages Only
**Four Groups of 16
***Four Groups of 16 Plus 80 Non-Group Users

In addition to normal termination of polling, an automatic repoll capability has been provided to allow continuous monitoring of reportback status activity. Also, an interrupt-and-resume feature provides a standby mode for polling operations to accommodate higher priority communications without the need to reinitiate a repoll of net members who have already reported back.

4.1.2 Remote Entry Automatic Check-in

STATE OF THE PROPERTY OF THE P

The microcomputer software provides an automatic remote self check-in before the start of actual polling operations. This allows potential users to become polling network members without manual intervention by the NCS terminal operator. A non-group polling list can be rapidly structured yet the work-load imposed on the operator by the large number of possible random addressees is minimized. This enhancement also allows check-in table entry while current polling operations are in progress.

The NCS terminal operator retains positive control of all check-ins, including remote check-ins with the following software features:

- The remote check-in mode can be enabled or disabled separately by COMSUP command.
- Check-ins can be verified using COMSUP commands to obtain hardcopy poll table printouts of listed poll check-ins and poll net members. (This includes both group and non-group check-ins.) Check-in addition/deletion can be accommodated by COMSUP command as needed.
- 3. Automatic slot assignment and reportback prioritization (order of reportback) of poll net members is done in the polling software before the start of actual polling operations; reassignment of slots is also possible with a COMSUP command.
- 4. An ALL CALL message initiated via COMSUP command by the NCS terminal operator informs all listed polling net members of their TDM slot assignments (to be dialed manually into their respective AFSATCOM synchronizers). Dial-up of respective slot assignments avoids subsequent contention problems during TDM polling operations should that mode become active.

4.1.3 TDM Auto Polling

This enhancement provides TDM auto polling in addition to the polling capabilities available in the non-TDM mode. Thus, AFSATCOM TDM-1 and TDM-2 system modes are made compatible with the new terminal polling capabilities. This implementation requires the AFSATCOM synchronizer in the CP terminal to act as a master (i.e., a net control element) during TDM polling operations.

To carry out auto polling in the TDM modes, the software utilizes a tabular listing containing the addresses of all terminals in the poll network and their respective assigned reportback time slots. The TDM mode provides 60 slots per frame. During actual polling operation in the TDM mode, eight contiguous slots containing up to six poll messages per slot at the beginning of each AFSATCOM frame are dedicated to sending polling messages from the CP terminal to the pollees. Three unassigned slots immediately following these eight may be used for break-in by unassigned or emergency users. The 48 slots following can then be used for terminal reportbacks. The final slot in the frame, however, is normally reserved for synchronization messages. Because each AFSATCOM terminal is assigned its own unique address, several pollees can be assigned to the same reportback slot. The microcomputer software presently provides for polling a maximum of 144 users in three successive frames (with 48 reportbacks per frame). This, however, is not an ultimate AFSATCOM System limitation. Up to three addressees may be assigned to the same slot under this polling scheme, but this number could be increased to accommodate additional users as necessary, if more frames are used.

The AFSATCOM TDM response of force terminals (or CP terminals in the slave mode) limits polling responses by a polled terminal to 40 message characters during a frame. Also, TDM polling does not have an encryption capability as presently implemented. However, various priority traffic indicators in the poll response message cause a printout alert message at the NCS terminal for subsequent operator action. These are described in more detail below.

4.1.4 Automatic Slot Assignment

In conjunction with TDM polling, the microcomputer software provides automatic slot assignment of listed poll net members for subsequent TDM reportback. Before actual start of polling, members of the network are informed of their slot assignments by means of an ALL CALL message generated by the software. This message also indicates which users have been accepted into the poll listing along with their priority order of reportback. The ALL CALL message, like

the slot assignment made during the course of compiling the poll table, is generated automatically by operator-initiated COMSUP command.

4.1.5 Encrypted Non-TDM Polling

The present AFSATCOM polling time-out limitation of 30 seconds precludes the use of long encrypted reportback sequences. The microcomputer system software, however, permits encrypted polling responses in the non-TDM mode of operation. The software is configured to recognize KG-35 encryption device message indicator (MI) sequences at the beginning of a message and automatically disables odd-parity checking in the AFSATCOM NB modem upon recognition of a valid MI header. While message traffic is being received during polling, no message time-out occurs in order to accommodate encrypted sequences of long lengths. Once received, the AFSATCOM MPU routes the incoming encrypted reportback to a KG-35 device for eventual decryption.

4.1.6 Error Detection and Correction (EDAC)

The second secon

The present AFSATCOM polling capability does not provide for any EDAC during either polled or non-polled operations. The microcomputer software does provide limited EDAC during polling through satellite downlink monitoring of poll messages. Polling message retransmission then takes place upon detection of error. This technique, however, requires full-duplex operation (simultaneous transmit and receive) whereas the demonstration hardware is interfaced with either the AFSATCOM OW or the 1x1 NB modem, both of which are configured as half-duplex equipment.

With a full-duplex channel, after transmission of a polling message during non-TDM mode, the microcomputer compares the message sent with the one received on the satellite downlink and causes a retransmission of the same message upon detection of any mismatch. A maximum of three automatic poll message retransmissions is possible with the present microcomputer software. No EDAC is presently provided in the TDM polling mode nor is EDAC provided on reportback responses, although no inherent limitation precludes such additional capability.

4.1.7 Variable Non-Reportback Time-out

A variable non-reportback time-out feature using a COMSUP command with the microcomputer software provides the terminal NCS operator with an added dimension for control. By controlling the time-out delay from the possible non-reportback of polled terminals, the operator can minimize the time-to-completion of polling the overall network. The operator enters the COMSUP command with the desired time-out duration expressed in seconds. Note that the present AFSATCOM non-reportback poll time-out is a non-changeable 30 seconds, whereas the microcomputer software allows for a 1 to 99 second range. Choice of the optimum delay depends on actual satellite and AFSATCOM I/O device delays and the polling strategy to be employed. The ability to vary the time-out delay can have profound effects on the overall polling completion time.

4.2 NEW COMMANDS AND MESSAGES FOR POLLING

The following new polling commands and message formats are possible with the microcomputer software.

4.2.1 COMSUP Commands

Table 4-2 summarizes the new COMSUP commands available to the NCS terminal operator when using the microcomputer system polling enhancements. Note that all the commands listed contain the ZNR character field in the first three character positions in order to identify this properly as an unencrypted message in an AFSATCOM preproduction Type 12 terminal. (AFSATCOM production-type terminals would require a "UUU" sequence.) The fourth character position is an exclamation point ("!") which signifies a COMSUP message to the microcomputer's improved polling software. The fifth character position of a COMSUP message is always a mode character. Characters six through nine are microcomputer address characters. (The microcomputer address is also changeable via a COMSUP command.) COMSUP commands intended for the microcomputer, and any other unencrypted messages, are input to the AFSATCOM terminal and intercepted by the polling software. These COMSUP commands are never transmitted to the AFSATCOM modem.

The seven message categories defining the new COMSUP commands are shown in table 4-2. These functional categories satisfy the increased control and data entry requirements for the various polling mode enhancements.

Table 4-2
COMSUP Commands For Polling Enhancements

Command Type	Function(s)	Command Message Format
Check-in Table Entry	Enable External Check-ins	ZNR!SABC1
	Disable External Check-ins	ZNR!SABCØ
	Single Priority Check-in	ZNR!CABCAAA1
	Single Routine Check-in	ZNR!DABCBBB1
	Single No-Traffic Check-in	ZNR!EABCCCCØ
	Group Check-in	ZNR!BABCDD1
	Priority Table Load	ZNR!1ABC ADR1 ADR2
	Routine Table Load	ZNR!2ABC ADR3 ADR4
	No Traffic Table Lo	ad ZNR!3ABC ADR5 ADR6
Check-in Table Deletion	Priority Table Re-initialize	ZNR!FABC
	Routine Table Re-initialize	ZNR!GABC
	No-Traffic Table Re-initialize	ZNR!HABC
	Group Table Re- initialize	ZNR!AABC
	Single Priority Deletion	ZNR!NABCPØ1
	Single Routine Deletion	ZNR!NABCRØ1
	Single No-Traffic Deletion	ZNR! NABCNØ1
	Single Group Deletion	ZNR!NABCGØ1

Table 4-2 (Continued)

COMSUP Commands For Polling Enhancements

Command Type	Function(s)	Command Message Format
Check-in/Poll Table Printout	Priority Table Printout	ZNR!UABCPXYZ
	Routine Table Printout	ZNR!UABCRXYZ
	No-Traffic Table Printout	ZNR!UABCQXYZ
	Group Table Printout	ZNR!UABCOXYZ
	Build/Printout of Poll Table	ZNR!MABC
	Poll Table Printout (Local)	ZNR!UABCZXYZL
	Poll Table Print- out (Local and	745 1445 674475
	Remote All Call)	ZNR!UABCZXYZR
External Table Transfer	Priority Check-in Table Transfer	ZNR!TABCXXYZ
	Routine Check-in Table Transfer	ZNR!TABCYXYZ
	No-Traffic Check-in Table Transfer	ZNR!TABCZXYZ
	Poll Table Transfer	ZNR!TABCTXYZ

Table 4-2 (Continued)

COMSUP Commands For Polling Enhancements

Command Type	Function(s)	Command Message Format
Poll Mode Control	Single Poll Enable	ZNR!JABC1S
	Single Poll Disable	ZNR!JABCØS
	Multiple Poll Enable	ZNR!JABC1M
	Multiple Poll Disable	ZNR!JABCØM
	Non-TDM Poll Interrupt	ZNR!LABC1Ø
	TDM Poll Interrupt	ZNR!LABCØ1
	TDM and Non-TDM Poll Interrupt	ZNR!LABC11
	Disable All Poll Interrupts	ZNR!LABCØØ
	EDAC Mode Fnable	ZNR!PABC1
	EDAC Mode Disable	ZNR!PABCØ
	Poll Message Time Delay	ZNR!KABC1Ø
	Reportback/Poll Message Slot Reassignment	ZNR! 0ABC10182634425004

Table 4-2 (Continued)

COMSUP Commands For Polling Enhancements

Command Type	Function(s)	Command Message Format
Miscellaneous	Disable COMSUP Msg. Printout	ZNR!4ABCØ
	Enable COMSUP Msg. Printout	ZNR!4ABC1
	Disregard All Non- COMSUP Msgs.	ZNR!VABC2
	Disregard Non-COMSU Msgs. when Polling	P ZNR!VABC1
	Enable Recognition of All Types of Msgs.	of ZNR!VABCØ
	RS-232 1/0 Enable	ZNR!QABC1
	RS-232 I/O Disable	ZNR!QABCØ
	Microcomputer Addres	SS ZNR!IABCCAB
	Program Re-initializ	ze ZNR!RABC
	Loop-Around Message	ZNR*
		Transmitted Message

4.2.2 Error Messages

Table 4-3 summarizes the new error messages available to the terminal operator with the microcomputer software. These messages assist in either flagging anomalous COMSUP command inputs to the microcomputer or indicating to the operator when pre-determined limits have been reached for polling check-in messages.

4.2.3 External Message Formats

Table 4-4 summarizes the types of externally generated messages from the AFSATCOM modem which are used for polling operations with the microcomputer software. These messages are divided into three basic categories:

- 1. Single Check-in
- 2. Check-in Table Transfer
- 3. Reportback

The single check-in message is normally sent by a terminal desiring to become a member of a poll network. It is accepted by the microcomputer software after the check-in mode has been enabled by the NCS terminal operator.

Check-in table transfer message formats allow entry of entire sets of check-ins involving multiple addressees into appropriately prioritized check-in tables.

Reportback message formats are normally sent by a terminal responding to a poll inquiry message. When the microcomputer software recognizes these reportback formats during polling operation, appropriate prioritization header messages are appended to the incoming message before printout on the terminal I/O device.

Table 4-3
Error Message Formats

Message Name	<u>Function</u>	Message Text
ADRERRORHDR	Informs of incorrect address characters	Incorrect ADR Characters
ERRORHDR	Informs of incorrect mode character	Incorrect Mode Character
TFRERRORHDR	Informs of attempted check-in table transfer with no check-ins listed	Table Transfer Error: No Check-ins
OVERFLOWHDR	Informs of Check-in limit being reached	Check-ins at the Limit

Table 4-4
External Message Formats

Message Type	Function		Format
Single Check-in	Priority Check-in	SOH C	ABC ADR1 Text
	Routine Check-in	SOH D	ABC ADR2 Text
	No-Traffic Check-in	SOH E	ABC ADR3 Text
Check-in Table Transfer	Priority Check-in Table Transfer	SOH X	ABC ADR1ADR2
	Routine Check-in Table Transfer	SOH Y	ABC ADR3ADR4
	No-Traffic Check-in Table Transfer	SOH Z	ABC ADR5ADR6
Reportback Message	Priority Report back	SOH P	ABC Text
	Routine Report back	SOH R	ABC Text
	No-Traffic Report back	SOH N	ABC Text

SECTION 5

MICROCOMPUTER HARDWARE

The microcomputer developed for demonstrating AFSATCOM polling improvements is shown in the functional block diagram in figure 5-1. It consists of an Intel SBC 80/20 Single Board Computer, a National Semiconductor Model BLC 416 16K-byte programmable read-only memory (PROM) board, an Intel Model 450 16K-byte random access memory (RAM) board, a power supply subsystem, and interface/display logic circuitry. The SBC 80/20, RAM, and eraseable PROM (EPROM) boards are housed in the Intel SBC 604 Modular Cardcage/Backplane which provides interconnection for up to four plug-in boards. The SBC 604 also allows interconnection of two or more cardcage backplane assemblies for expansion, in addition to power supply connectors and signal line termination circuits. A small table-top 19-inch modular rack mounting, shown in figure 5-2, contains the cardcage, along with the power supply subsystem, slot selection and display circuits, bypass switches, test points, and cable terminations for circuit connections to the AFSATCOM System.

5.1 INTEL SBC 80/20 SINGLE BOARD COMPUTER

The Intel SBC 80/20 Single Board Computer is the heart of the microcomputer implementation for polling improvements. It is one of Intel's line of computer products and takes full advantage of large scale integration (LSI) technology to provide a self-contained computer capability on a single 6.75 x 12.00 inch printed circuit board. On-board capability provides a central processing unit (CPU), system clock, read/write memory, nonvolatile read-only memory (ROM), I/O ports and drivers, a serial RS-232 communications interface, priority interrupt logic, and two programmable timers. In addition, on-board multibus control logic and bus expansion drivers allow interfaces to multibus-compatible boards including expansion memory, digital and analog I/O expansion boards, peripheral controller, and other single board computers. A complete functional description of the SBC 80/20 is found in reference 1. Details of the SBC 80/20 are found in reference 2. Table 5-1 lists integrated circuit (IC) locations for polling program storage on the SBC 80/20.

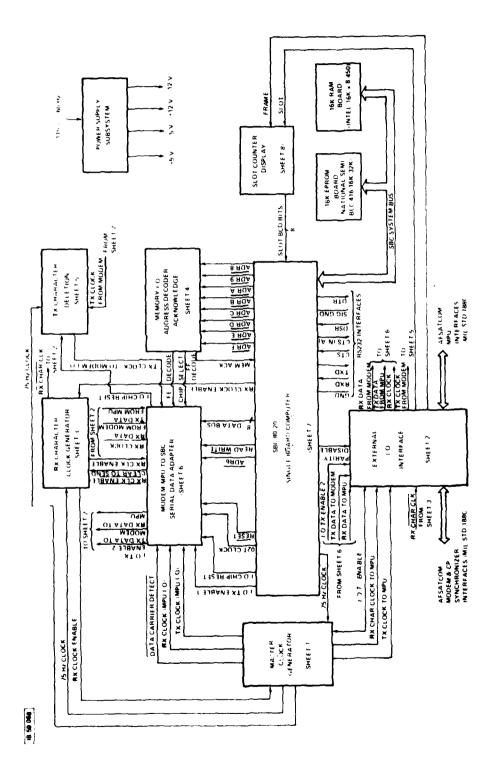
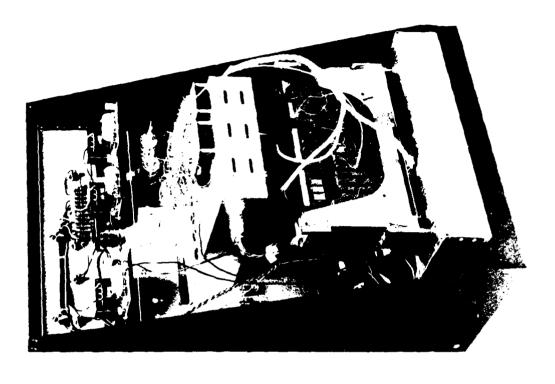


Figure 5-1 Microcomputer Functional Block Diagram



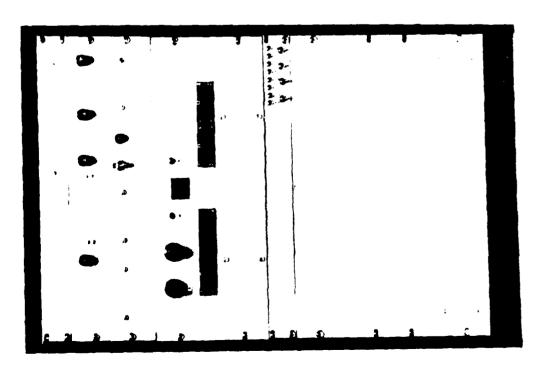


Figure 5-2 Microcomputer System Assembly

Table 5-1
EPROM (2708) Location Summary

Prom #	<u>1C</u>	Board	Start ADR (Hex)	Finish ADR (Hex)
1	A79	SBC 80/20	0000	03FF
2	A64	SBC 80/20	0400	07FF
3	A51	SBC 80/20	0800	OBFF
4	A37	SBC 80/20	0000	OFFF
5	U33	BLC 416	1000	13FF
6	U21	BLC 416	1400	17FF
7	U9	BLC 416	1800	1BFF
8	U3	BLC 416	1000	1FFF
9	U34	BLC 416	2000	23FF
10	U22	BLC 416	2400	27FF
11	Ulo	BLC 416	2800	2BFF
12	U4	BLC 416	2000	2FFF
13	U35	BLC 416	3000	33FF
14	U23	BLC 416	3400	37FF
15	U11	BLC 416	3800	3BFF
16	U5	BLC 416	3000	3FFF

5.2 NATIONAL SEMICONDUCTOR BLC 416 16K PROM BOARD

The National Semiconductor BLC 416 16K PROM expansion board provides sockets for up to 16K bytes of 2708 EPROMs (32K bytes for 2716 EPROMs). For the polling program, however, only 12 of the 16 2708 EPROMs required for program storage actually reside on this board; the four remaining 2708 EPROMs reside on the SBC 80/20. Table 5-1 summarizes the physical locations of the EPROM program storage for execution of the polling software. Switches to either enable or disable various memory blocks are provided on this board along with jumper-switch selectable addresses for each 4K block (8K when used with the 2716 EPROMs). This allows independent selection of base addresses of individual memory blocks on 4K byte boundaries. Along with on-board programming, the BLC 416 provides multibuscompatible address, data, and command signals. Table 5-1 lists the IC locations for polling program storage on the BLC 416 PROM board.

5.3 INTEL 450 16K RAM BOARD

The Intel 450 16K RAM is a multibus-compatible board which contains 16K bytes of read/write memory and uses 2107C dynamic memory components with on-board refresh circuitry for all the dynamic memory elements. Read/write buffering, which also resides on-board, buffers all data written into or read from the memory array and includes a jumper-selectable starting address for 16K contiguous addresses at 16K boundaries. For the polling application, the lowest RAM actress is set at 4000H and the highest RAM address is 7FFFH.

5.4 I/O AND DISPLAY CIRCUITS

Figure 5-3 shows the additional cabling needed in the Type 12 CP terminal for interconnecting the I/O and display boards of the microcomputer system with the AFSATCOM System. No changes to existing AFSATCOM interfaces were made; only cabling and connector additions were employed. Each of the I/O and display circuits is detailed below.

5.4.1 Master Clock Generator

Figure 5-4 shows the Master Clock Generator circuit. This circuit provides various gated 75 Hz clocks and associated signals for both the transmit and receive functions utilizing a 75 Hz reference input from one of two timer circuits on the SBC 80/20.

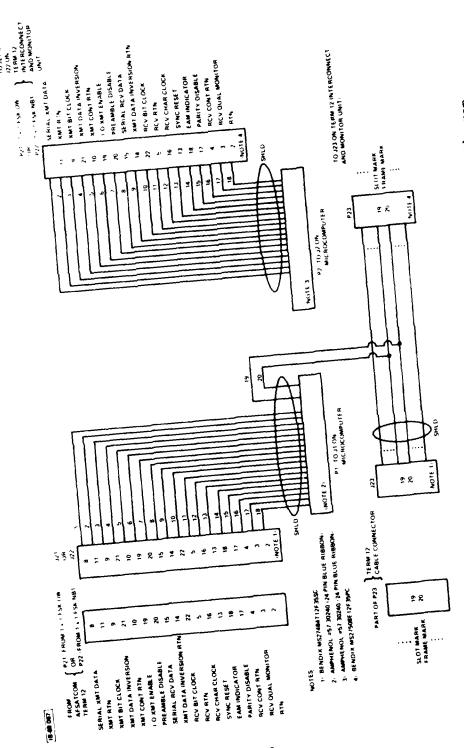
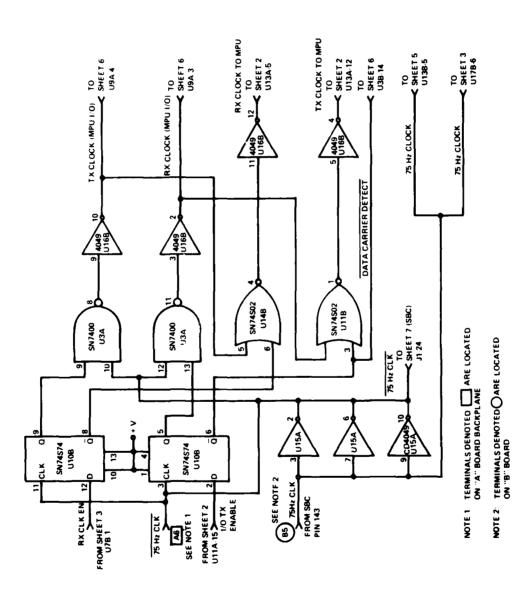


Figure 5-3 AFSATCOM T12-to-Microcomputer System Interface Diagram



THE RESERVE OF THE PARTY OF THE

18 56 066

Figure 5-4 Master Clock Generator (Sheet 1)

5.4.2 External I/O Interface

Figure 5-5 shows the external I/O interface circuits. This circuitry provides the level conversion necessary to connect the MIL-STD-188C interfaces found in the AFSATCOM Type 12 terminal with the complementary metal-oxide semiconductor (CMOS) and transistor-transistor logic (TTL)-compatible circuits of the microcomputer system.

5.4.3 RX Character Clock Generator

Figure 5-6 shows the RX character clock generator circuit. This circuitry provides synchronous receive character clock for the MPU receive data interface normally supplied by the AFSATCOM Type 12 terminal. The microcomputer system does not use the RX character clock signal supplied by the AFSATCOM modem. However, for proper AFSATCOM operation, this clock is reconstituted and supplied to the MPU input circuitry.

5.4.4 Memory I/O Address Decoder/Acknowledge

Figure 5-7 shows the memory I/O address decoder/acknowledge circuit. Its primary function is to generate select signals for memory address decoding for a pair of Motorola MC6852 serial data adapter ICs. These select signals, along with the memory acknowledge signal, permit the ICs to be utilized as if they were an inherent part of the SBC 80/20 memory addressing space. Read or write operations to these devices then use the memory addresses which serve these two ICs.

5.4.5 TX Character Deletion

The TX character deletion circuit, shown in figure 5-8, deletes two redundant message characters encountered with the MC6852 IC used in conjunction with the AFSATCOM modem transmit interface. Actual transmit character deletion takes place before the start of transmission to the modem and occurs only during the time interval when the AFSATCOM modem is in the process of transmitting its own "W U SYN SYN" preamble.

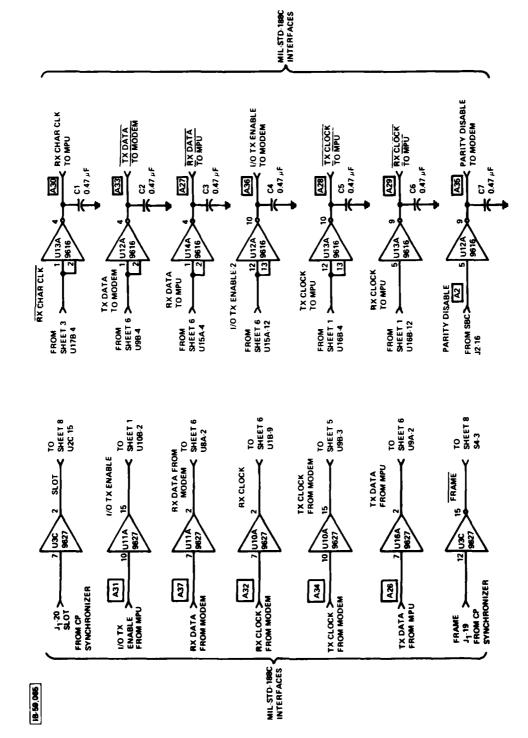


Figure 5-5 External I/O Interface (Sheet 2)

いんしてい あいましたました かんかん

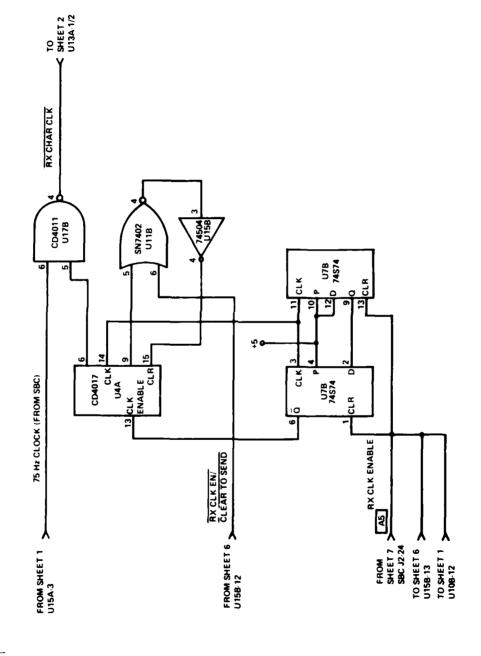


Figure 5-6 RX Character Clock Generator (Sheet 3)

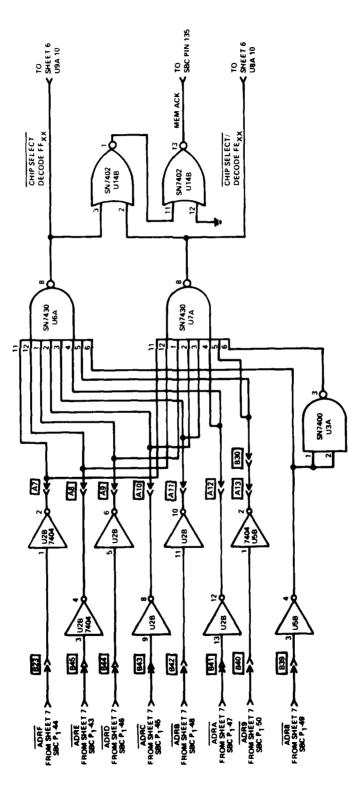


Figure 5-7 Memory I/O Address Decoder/Acknowledge (Sheet 4)

the second secon

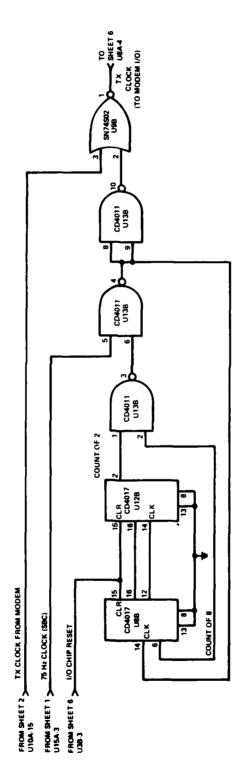


Figure 5-8 TX Character Deletion (Sheet 5)

5.4.6 Modem/MPU-to-SBC Serial Data Adapter

Figure 5-9 shows the modem/MPU-to-SBC serial data adapter circuit. Its primary function is to transfer serial data to/from the AFSATCOM modem or MPU to/from the SBC 80/20. The SBC 80/20 reads or writes these data as if they were located in a memory address. Selection of either the modem I/O or MPU I/O takes place via a register select line with read/write operations controlled separately.

The heart of this circuitry consists of a pair of Motorola MC6852 programmable synchronous serial data adapter (SSDA) ICs. Each provides a three-character buffered bi-directional serial interface for synchronous data exchange. The MC6852 design incorporates bus interface logic to allow parallel data transfer over the SBC 80/20 bi-directional parallel data bus. The actual configuration of the SSDA is programmed via the data bus using the polling program system initialization software. SSDA internal programmable control registers provide word length, transmit, receive, synchronization, and interrupt controls. Status, timing, and other SSDA control lines provide additional peripheral or modem functions.

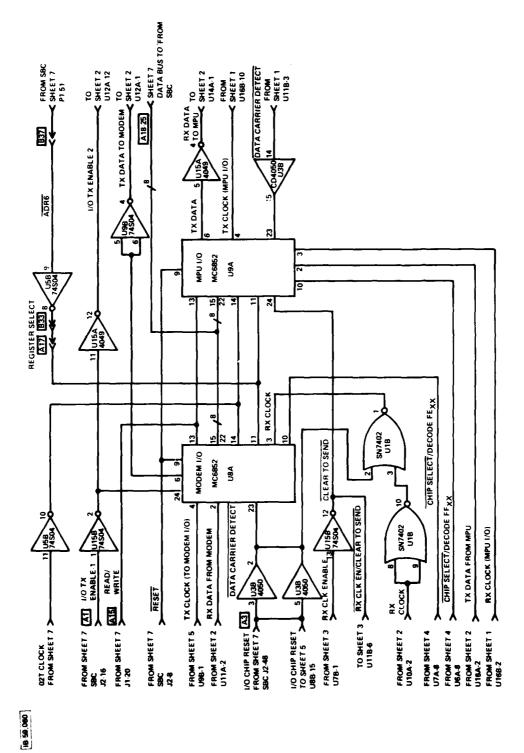
5.4.7 SBC 80/20 Interfaces

Figure 5-10 summarizes the I/O and display interfaces within the microcomputer system. Jumpers and parallel terminations employed, along with physical board modifications for the SBC 80/20, are indicated.

5.4.8 Slot Counter/Display

The second of th

Figure 5-11 shows the slot counter/display circuit interfaces to the SBC 80/20 and the I/O circuits. Slot pulses from the AFSATCOM synchronizer are used to increment the slot display on a pair of Texas Instruments TIL 306 ICs while the frame pulse from the synchronizer (or a manual toggle switch) provides a reset function back to zero slot indication. A pair of four-bit binary coded decimal (BCD) output lines (8 parallel lines, total) to one of the SBC 80/20 parallel input ports is used by the polling software in calculating the current AFSATCOM slot. As implemented in the demonstration system software, zero-slot indications signal the non-TDM AFSATCOM mode and non-zero indications signal TDM operation.



はないのから からまるとなる 一般の 日本の おいかい こうしゅ 日本の 日本の 日本の こうしょう

Figure 5-9 Modem/MPU-to-SBC Serial Data Adapter (Sheet 6)

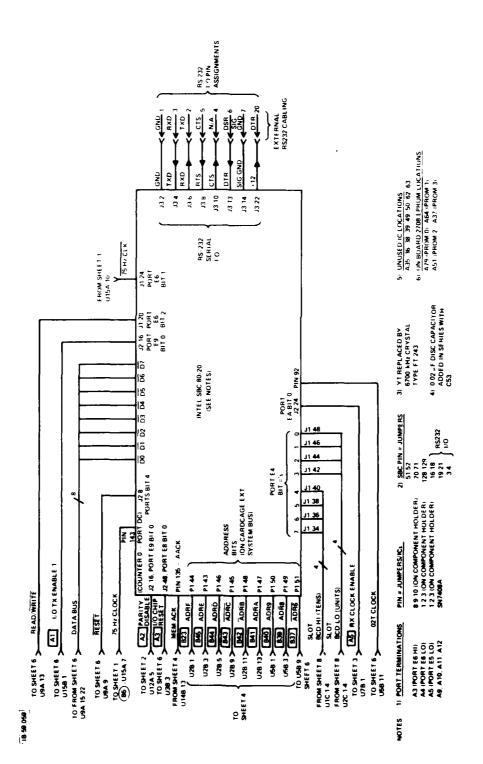


Figure 5-10 SBC 80/20 Single Board Computer Interfaces (Sheet 7)

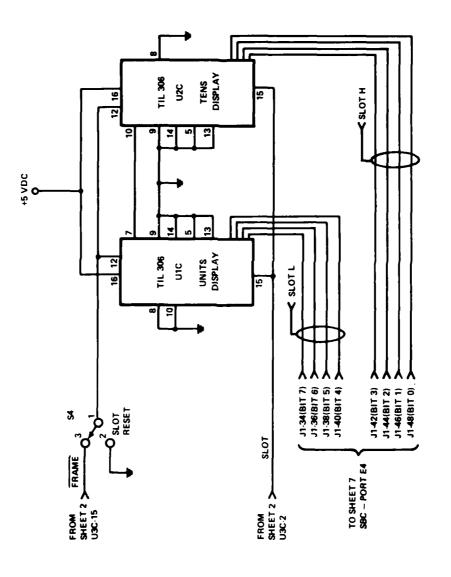


Figure 5-11 Slot Counter/Display (Sheet 8)

18-50.063

5.4.9 RS-232 Serial Interfaces

Figure 5-12 summarizes the cable interfaces employed with the microcomputer system and made possible by serial RS-232 interface circuitry which employs a software-programmable 8251 USART IC on the SBC 80/20 card. Shown are cable interfaces for use with a Bell 103J modem, a "NULL" modem, a Lear Siegler ADM-3A CRT terminal, and a Texas Instruments Model 765 Intelligent Terminal.

When the polling software is commanded into its RS-232 mode, all messages normally routed to the AFSATCOM MPU interface are rerouted to the RS-232 serial interface port at a 300 b/s rate (instead of 75 b/s). Messages destined for transmission to the AFSATCOM modem or COMSUP commands normally originating at an AFSATCOM ASR device may now be entered into this RS-232 interface. While in this mode, the AFSATCOM ASR may still be used for all transmitting functions; however normal reception from the OW (or NB-1) channel will be diverted to this RS-232 port. When not in the RS-232 mode, normal NB interfaces to the MPU are utilized.

5.5 MODULAR RACK HARDWARE FEATURES

When the microcomputer rack assembly is installed in the Type 12 terminal, two multi-pole bypass switches mounted on the front panel can disengage the microcomputer system from the MPU, NB modem, and synchronizer interfaces without physical removal of connectors. Connector terminations on the rear of the rack provide convenient access for cables to the AFSATCOM Type 12 CP terminal.

The modular rack assembly is designed for ease in removing all mounted components and subsystems. Test points mounted on the front panel of the rack also provide a convenient access to all interfaces for ease in trouble-shooting. Front panel fuses on the rack serve to protect the various power supplies in the power supply subsystem. An on-off power line switch and a line voltage indicator are below these fuses.

Finally, a slot display reset switch and eight BCD slot selection switches are also front-panel mounted for convenient control of polling test functions related to TDM and non-TDM modes.

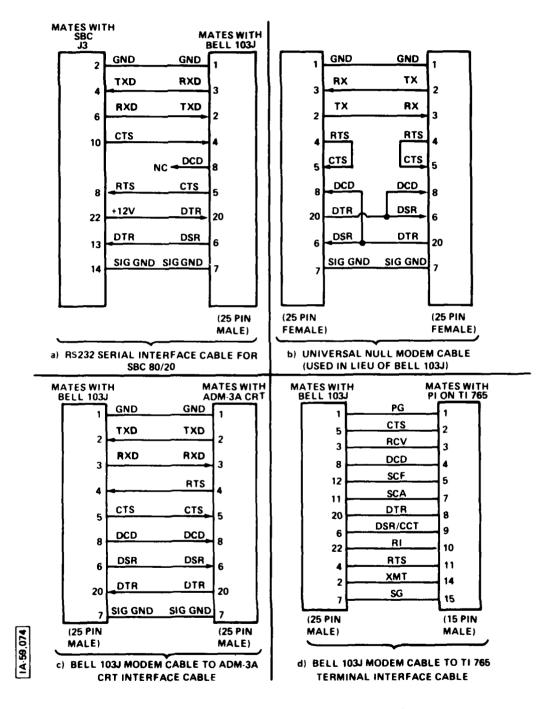


Figure 5-12 RS-232 Serial Interfaces

SECTION 6

MICROCOMPUTER SOFTWARE

Figure 6-1 shows the major elements of the microcomputer facility at MITRE-Bedford which was used for overall software development for the polling improvement project. The Tektronix 8002A software development system and the MITRE Time Sharing Option (TSO) system assets related to polling software development are described below.

The microcomputer software for the improved polling demonstration capability was written in PL/M-80 high-level language on TSO. The PL/M-80 source coding, which is essentially self-documenting, takes advantage of modular interrupt-free, structured programming techniques and attempts to minimize (but not entirely eliminate) interactions between the various modular segments of the overall program.

After appropriate compilation and debug on TSO, followed by downloading to the Tektronix 8002A microprocessor laboratory, executable code was installed in the microcomputer system using 2708 EPROM non-volatile memory ICs. Execution of this code in the microcomputer system provides the processing functions needed to satisfy the polling enhancements described in section 4.

6.1 FLOW CHARTS

The polling software for the microcomputer system is functionally described by the flow chart of figure 6-2. Following power turn-on of the microcomputer system, the software executes the Initialization Sequence, followed by the main program loop starting at the Slot Calculation Algorithm and ending at the TX Output Algorithm. Return to the Initialization Sequence is also possible by means of a software re-initialization COMSUP command. Unless the software is re-initialized, however, normal program execution continues in this endless loop. The time-to-completion of a single cycle within this loop is typically less than a single character interval at the AFSATCOM 75 b/s rate. Coupled with this capability is a three-character buffer in the serial synchronous MPU and modem interfaces employed for use with the AFSATCOM hardware (see subsection 5.4.6) which avoids loss of data during program execution in an interrupt-free manner.

The second of th

さん かんしゅう かんかん かんかん

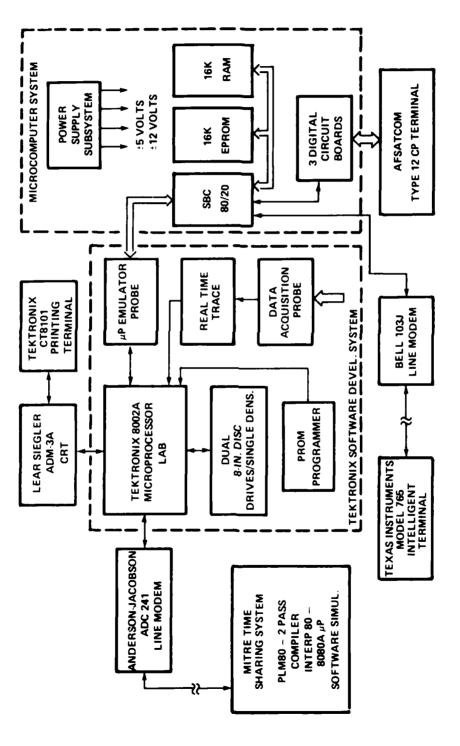
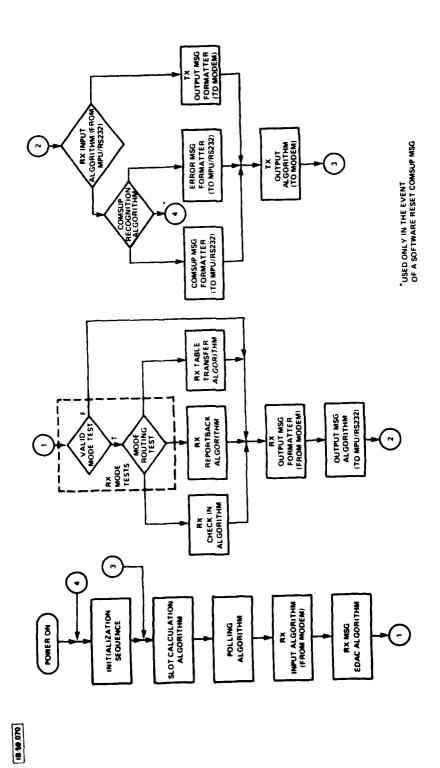


Figure 6-1 Microcomputer Development Facility



THE RESIDENCE OF THE PARTY OF T

Figure 6-2 Polling Software Program Flowchart

Table 6-1 describes the various program modules along with their major performance characteristics during program execution. The polling algorithm is the heart of the overall polling program software; a more detailed flow chart of this polling algorithm is provided in figure 6-3.

6.2 PL/M-80 SOURCE CODE AND MEMORY ALLOCATIONS

Table 6-2 contains the improved polling software symbol table and memory map which identifies RAM and ROM (EPROM) label locations, variables, and constants used throughout the software. This table is consistent with the program module PL/M-80 labels shown in table 6-1 and the address locations in table 5-1 so that the start of each major program module may be readily identified. This also allows ease in modifying or upgrading existing software by appropriate EPROM replacement within the microcomputer system. Appendix B contains the actual PL/M-80 source code which was written on TSO.

6.3 MITRE TSO SYSTEM PL/M-80 SUPPORT

PL/M-80 assets available on TSO which were utilized for the microcomputer software development are described below.

6.3.1 PL/M-80 High Level Language

PL/M-80 is an advanced high level programming language available on the MITRE TSO system. Specifically designed to simplify the job of system programming for the Intel 8080 8-bit microprocessor, it provides a very effective software development and maintenance tool, well suited to the requirements of the microcomputer system designer. PL/M-80 is also easy to learn and facilitates rapid program development and debugging since it is an algorithmic language in which the program consists of a sequence of declarations and executable statements naturally expressing the algorithm to be performed. Thus, the programmer is free to concentrate on system development through use of modern structured programming techniques at a high level, rather than dealing with assembly language details (such as register allocation, etc.). For a complete description of PL/M-80 see reference 3.

6.3.2 PL/M-80 2-Pass Cross-Compiler

The second secon

The 8080~PL/M cross-compiler consists of two distinct programs which must be executed consecutively to perform a complete

Table 6-1

THE RESERVE OF THE PROPERTY OF THE PARTY OF

Program Software Module Description

Module Initialization Sequence	Label(s) Program\$Start	Eunctions Defines I/O port parameters, declares/initializes constants/variables; initializes key memory locations; initializes SBC programmable counters: defines I/O
Slot Calculation Algorithm	IPOLL 1	memory locations. Calculates current slot; sets TDM mode flag; converts BCD input from slot display to decimal using conversion algorithm.
Polling Algorithm	POLLSXMIT	Checks poll mode flag; checks for polling completion; checks for TDM or non-TDM polling mode; checks for pollinterrupt; signals completion of polling; updates EDAC table entry during non-TDM polling.
RX Input Algorithm	IPOLL II	Tests RX input buffer character status of 1/0 chip; fetches RX character when available; tests for last character received state; tests for RX buffer memory overflow; tests for encrypted mode initialization sequence; tests mode/address characters for subsequent processing sequence (sets MPU job flag).
EDAC Algorithm	EDAC\$ROUTINE	Provides EDAC for polling messages transmitted via downlink RX monitoring. (If EDAC criteria are satisfied, sets retransmit flag for a non-TDM mode polling message for up to three retransmissions.)
RX Mode Tests	MJT2	Tests RX mode characters from modem for check-ins, reportbacks, and table transfers; provides routing to various RX mode algorithms.

Table 6-1 (Continued)

THE REPORT OF THE PROPERTY OF

Program Software Module Description

Functions	Identifies invalid check-ins from modem; logs valid check-ins into appropriate check-in tables; composes output message for MPU with appropriate check-in headers; formats overflow message when check-ins are at limit for MPU (and also for modem when poll mode is not active); automatically cancels check-in mode when checkins are at limit.	Composes output message to MPU with appropriate reportback header when a valid reportback is received from modem.	Identifies invalid table transfers received from the modem; logs valid table transfers into appropriate check-in tables; composes output messages to MPU with appropriate table transfer headers; formats overflow messages when check-ins are at limit for MPU (and also for modem when poll mode is not active); automatically cancels check-in mode when check-ins are at limit.	Checks if message delay is active; increments delay counter as required if msg delay is enabled; verifies need to output a message; checks availability of buffer space for outputting a character, outputs data to either MPU output (memory 1/0) or RS-232 1/0 port, depending on which is activated; shifts down internal storage buffer upon completion of message transmittal; enables clock output to MPU on start of message transmittal.
Label(s)	CS\$TEST	PRB\$LOAD RRB\$LOAD NRB\$LOAD	PTT\$LOAD RRB\$LOAD NTT\$LOAD	MPUŞTABLEŞBUILD
Module	RX Check-in Algorithm	RX Reportback Algorithm	RX Table XFER Algorithm	RX Output Msg Algorithm

Table 61 (Continued)

Program Software Module Description

Functions	Tests receive input buffer character status of RS-232 1/0 if activated; fetches character if available; tests receive input buffer character status of 1/0 chip; fetches character if available; tests for last character status; tests for buffer overflow; tests for valid commands and determines processing sequence by setting an output flag; determines when complete message has been received; verifies need to ignore all but command-type messages.	Checks for correct command-type message lengths; validates receipt of correct address characters; identi- fies various mode and address errors, checks for valid command-type mode characters; routes command-type message to appropriate command message algorithms.	Formats command-type messages to MPU input or RS-232 1/0 interface.	Formats appropriate error messages to MPU input or RS-232 1/0 interface.
Label(s)	1POLL101	COMPOSE 1	COMSUPLOAD1	MODEERROR1 ADRERROR1 OVERFLOW1 TFRERROR1
Module	RX Input Algorithm	COMSUP Recogni- tion Algorithm	COMSUP Msg Formatter (To MPU/RS-232)	Error Msg Formatter (To MPU/RS-232)

Table 6-1 (Concluded)

Program Software Module Description

Functions	Formats output messages to modem; deletes first three characters upon recognition of special character in position $\#4$; resets output flag for modem messages.	Checks if message delay is activated; increments appropriate delay counter as required if delay is active for TDM and non-TDM modes; verifies need to output a message; monitors state of receive input messages from modem to avoid transmitting while messages are received; signals modem when message is ready for transmission; shifts down internal storage buffer on completion of message transmittal.
Label(s)	MODEM\$OUTPUT\$- COMPOSE1 MODEM\$OUTPUT\$- COMPOSE2	0P0LL101
Module	TX Output Msg Formatter (To Modem)	TX Output

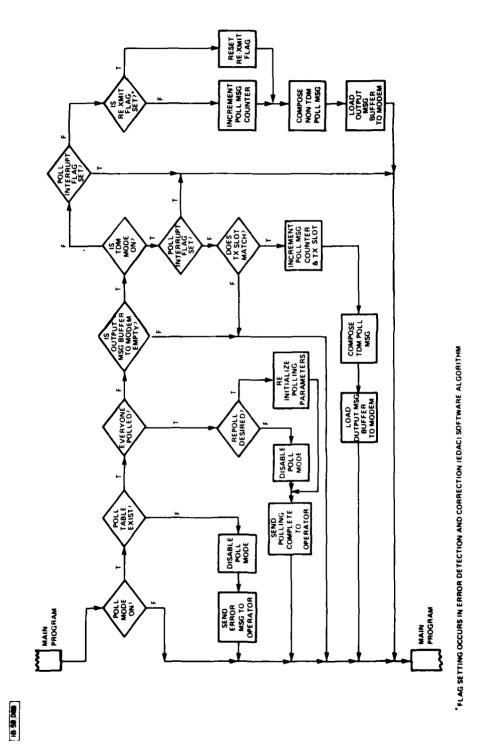


Figure 6-3 Polling Algorithm Flowchart

Table 6-2
Symbol Table and 'Memory Map

Name	Location (Hex)	Name	Location (Hex)
ADR1	4638	CLK3	4662
ADR2	4639	CLKCNT	5792
ADR3	463A	CLKCNT1	4664
ADRCHANGE	1B80	CLKCNT2	4666
ADRERROR1	1D67	CLKCNT3	4668
ADRERRORHDR	037C	CLKP	579A
AUTOPOLLRESTART	0611	CLKP1	465E
AUTOREPOLLFLAG	4684	CLKP2	4661
BUFFERDELAY	579C	CLKP3	4663
BUFFERDELAY1	466A	CMODE	6661
CC	6668	CMODECANCEL	0D99
CC1	5760	CMODECANCEL1	ODFC
CCPRESENT	4658	COMPOSE1	17FF
CC1PRESENT	465A	COMSUPLOAD1	296F
CCTEST	0B25	CS	5775
CHAR	5776	CS1	4688
CHAR1	4689	CS3	464A
CHAR3T	0998	CS4	464B
CHAR4T	09B1	CSDATAPOINTER	5764
CHAR4T1	166B	CS1DATAPOINTER	576C
CHAR5T	09CA	CSTEST	OB13
CHARSTORE	5D64	DELAY	579B
CHARSTORE1	468E	DELAY1	465F
CHECKINDELETE	247A	DELAYCAL	1D3A
CHECK INMODESET1	28D5	DELAYROUTINE	1442
CINUMTEST	0D76	DELAYROUTINE1	03D7
CLK	5799	EATMSG1	1773
CLK1	465D	EATMSGFLAG1	465C
CLK2	4660	EATMSGSET	1AF3

Table 6-2 (continued)

Name	Location (Hex)	Name	Location (Hex)
EDACFLAG	4656	INPUTS DATAPOINTER	5768
EDACMODESET	1B4A	INPUT103DATAPOINT	ER 5 76 E
EDACROUTINE	0A33	INPUT105DATAPOINT	ER 5770
ENC	5777	J	5762
ENCRYPT	0937	LCT	09E0
ENCTEST	0904	LCT1	1685
ENDLOAD	3833	LINECOUNT	468D
EOMTEST	13E0	LOADADR	37A1
EOMTEST1	3D3F	MEMORY	7400
ERRORHDR	039A	MJF	5778
GC	08BD	MJT1	093E
GC1	1614	MJT2	OAC7
GCIPOINTER	578C	MODE	5779
GCIRESET	2960	MODEERROR1	2A08
GCISTORE	5C23	MODEMIOCTL1POINTE	R 4632
GPOLLBUILD1	3C92	MODEMIOCTL2POINTE	k 4634
GT1	2278	MODEMOUTPUTCOMPOS	E1 1700
GTABLEBUILD1	3548	MODEMOUTPUTCOMPOS	E2 1783
GTABLEDUMP1	34C1	MPUCINUMTEST1	2824
GTDHEADER	0362	MPUIOCTL1POINTER	462E
I	6662	MPUIOCTL2POINTER	4630
11	575A	MPUTABLEBUILD	0E58
INITIALMSG	0461	MSGCOUNT	6534
IPLTEST	1915	MSGCOUNT1	4E5E
IPOLL1	04BF	MSGDELAY	579E
IPOLL11	087C	MSGOVERDELAY	4672
IPOLL101	15A8	N	577C
IPOLL1011	15D4	NC	5786
INPUT3DATAPOINTER	5766	NCIHEADER	02CD

Table 6-2 (continued)

Name	Location (Hex)	Name	Location (Hex)
NCIPOINTER	5790	OPOLL1	13BB
NCIRESET	2945	OPOLL101	3D1A
NCISTORE	5AA2	OUT1	468A
NCITEST	OCC8	OUTCHAR	148C
NOCOMSUPLOAD1	29FD	OUTCHAR1	3EA7
NONTDMDELAY1	3DF6	OUTCOUNT	6666
NONTDMDELAYSET	1CD2	OUTFLAG1	1680
NONTDMMSGDELAY1	466C	OUTPUTCOMPOSE	16CF
NONT DMPOLL	0622	OUTPUT1DATAPOINTER	576A
NPOLLBUILD1	3C12	OUTPUT101DATAPOINTE	R 5772
NRBHEADER	02E1	OUTTABLE	666B
NRBLOAD	0F9A	OUTTABLE1	4F8A
NT1	2201	OUTTABLEBUILD	0E64
NTABLEBUILD1	339В	OUTTABLE1STATUSTEST	05E9
NTABLEDUMP1	3314	OVERFLOW1	2847
NTDHEADER	0343	OVERFLOW2	284C
NTDMDELAYH	467C	OVERFLOWHDR	03D9
NTDMDELAYL	467E	PCIHEADER	0241
NTDMEDACTABLE	464F	PCIPOINTER	5784
NTDMPOLLI NTERRUPTFI	LAG 4686	PCIRESET	292D
NTTHEADER	02F5	PCISTORE	57A0
NTTLOAD	1278	PCITEST	OB43
NTTLOAD1	200A	POLLDUMPBUILD1	372F
NTTLOADEND	1348	POLLDUMPHDR	03F4
NTTR1	2CEB	POLLFLAG	4683
NUMGCI	577E	POLLINGMODETEST	05FF
immici	578E	POLLINTERRUPT	1C5B
MUMPCI	5732	POLLMODE	1BB4
MUMRCI	5783	POLLMODEERROR	0550

Table 6-2 (continued)

Name	Location (Hex)	Name	Location (Hex
POLLMSGCOMPLETE	0580	PTABLEBUILD1	3042
POLLMSGCOUNT	464C	PTABLEDUMP1	2FBB
POLLMSGOVER	0410	PT1COUNT1	73FE
POLLOVERTEST	0558	PTDHEADER	030A
POLLPOINTER1	73FC	PTTHEADER	02E5
POLLSLOT 1H	463B	PTTLOAD	OFF3
POLLSLOT 2H	463C	PTTLOAD1	1E3D
POLLSLOT 3H	463D	PTTLOADEND	1002
POLLSLOT 4H	463E	PTTR1	2845
POLLSLOT 5H	463F	QUIT	3FAB
POLLSLOT6H	4640	RCIHEADER	0209
POLLSLOTIL	4641	RCIPOINTER	578A
POLLSLOT 2L	4642	RCIRESET	293E
POLLSLOT 3L	4643	RCISTORE	5921
POLLSLOT4L	4644	RCITEST	0C05
POLLSLOT5L	4645	REPOLLFLAGCOUNT	4657
POLLSLOT6L	4646	REPOLLTEST	0572
POLLTABLE1	6E3B	RETRANSMITFLAG	4676
POLLTABLEBUILD1	3B34	RETRANSMITFLAGSET	OAAD
POLLTABLEDUMP1	36B1	RPOLLBUILD1	3B96
POLLTABLEDUMP 2	366E	RRBHEADER	O2DD
POLLTABLETX	3A3E	RRBLOAD	0F41
POLLTABLETXFLAG	4685	RS232FLAG	4637
POLLTTR1	2DBE	RS232GCI	161D
POLLXMIT	0530	RS232INPUT1	1589
PRBHEADER	02D1	RS232MODESET	IABD
PRBLOAD	OEE8	RS232OUTCHAR	14B1
PROGRAMSTART	0007	RS 232SPACETEST	1427
PT1	2114	RT1	218A

Table 6-2 (concluded)

Name	Location (Hex)	Name	Location (Hex)
RTABLEBUILD1	31EE	TCLK	466F
RTABLEDUMP1	3167	TCLK1	4673
RTDHEADER	0327	TCLKCNT	4671
RTTHEADER	02F1	TCLKCNT1	4675
RTTLOAD	1135	TCLKP	4670
RTTLOAD1	1F23	TCLKP1	4674
RTTLOADEND	1205	TDMDELAYROUTINE1	3DB3
RTTR1	2C18	TDMFLAG	4682
RXBUSYOVERRIDE1	3E60	TDMPOLL	0739
SHIFTDOWN	1408	TOMPOLLINTERRUPTFLA	G 4687
SHIFTDOWN1	3EC2	TDMPOLLTXSLOT	4677
SLOT	467A	TDMPOLLTXSLOTINITIA	L 4649
SLOTCOUNT	467B	TFRERROR1	2E7B
SLOTGUARDTIME	4680	TFRERRORHDR	03B8
SLOTH	4678	TOTALNUMCI	5780
SLOTL	4679	TTLOADEND1	20F1
SLOTREASSIGN	19AD	TTR1	2A94
SPACE	577B	TXSLOTINITIALH	4647
SPACE1	468C	TXSLOTINITIALL	4648
SPACETEST	1401	WAITING	577A
SPACETEST1	03D5	WAITING1	468B
ST	0916	WW	08A7
STARINSERT	0B1C	XX	089В
STARTUPMSG	042E	YY	15F2
TBLDUMP1	2EE1	2 7.	15FE
TC	6664		
TCl	575C		
TCINITIAL	5794		
TClinitial	5796		

compilation of a PL/M source program. The two programs are known as Pass 1 (PL/M-81) and Pass 2 (PL/M-82) of the PL/M compiler. They are written in ANSI standard Fortran IV and are installed on the MITRE TSO system.

The first pass of the compiler reads a PL/M source program and converts it to an intermediate form on work files. As an option, a listing of the input source program may be obtained during this pass. Errors in program syntax are detected at this stage and appropriate error messages are sent to the list file.

The second pass of the PL/M compiler processes the intermediate files created by Pass 1 and generates machine code. This machine code, which can be in either BNPF or hex format, may be loaded and executed on the SBC 80/20 microcomputer, on the Tektronix 8002A, or simulated using INTERP/80, a cross-simulator of the 8080 microprocessor unit. It may also be used for programming ROMs. Pass 2 of the compilation process can produce a symbol table and mnemonic listing of the generated machine code. Errors detected during this phase will be reported in the list file which is produced.

Figure 6-4 illustrates the overall file structure and flow of program execution of the PL/M cross-compiler available at MITRE. For a complete description of the PL/M-80 2-pass cross-compiler see reference 4.

The run-time organization of the memory storage allocation for a compiled PL/M-80 object program is shown in figure 6-5. Memory is allocated in three sections:

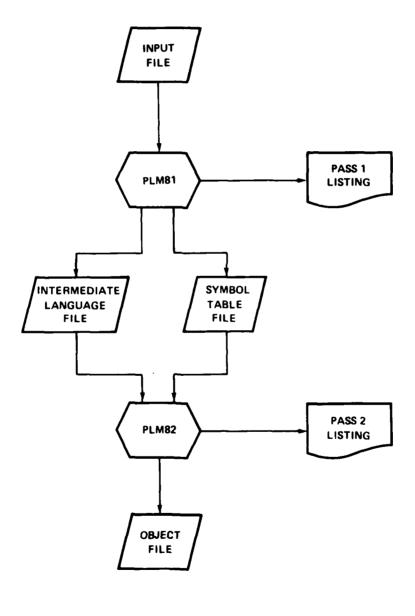
- 1. Instruction Storage Area (ISA).
- Variable Storage Area (VSA)
- 3. Free Storage Area (FSA)

さんしょう かんしゅう かんかん

The ISA is occupied by the machine code generated by the PL/M source and variables declared in DATA declarations.

The VSA is located above the ISA, and contains (in order of decreasing address):

- 1. Variables, other than DATA variables, declared in the PL/M source. They are arranged in order of declaration. ADDRESS variables are not aligned.
- 2. Compiler generated temporaries (i.e., workspace used in the object program, but not explicitly declared).



1A-59,076

Figure 6-4 File Structure and Flow of Program Execution on 8080 PL/M Cross Compiler

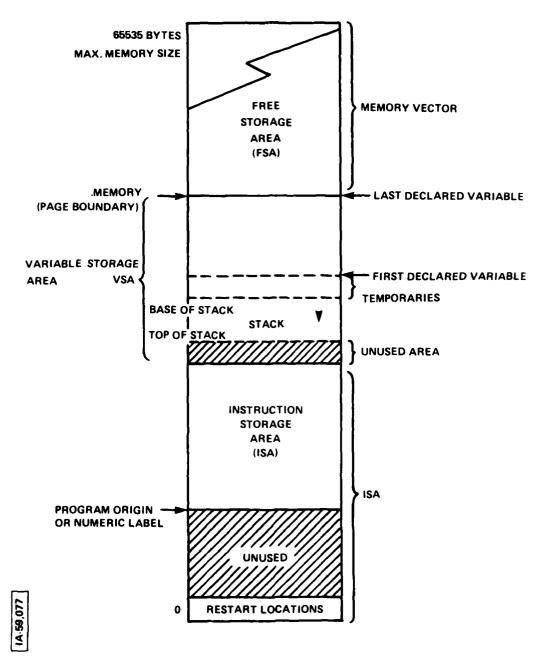


Figure 6-5 Run-Time Storage Organization of Memory Storage Allocation

3. The stack. The size of the stack area is determined by the compiler, unless explicit overrides are used.

The compiler will normally locate the VSA directly above the ISA. However, the compiler user may specify the first page of memory of the VSA explicitly. (A page of memory contains 256 bytes.) This may be used, for example, to ensure that the VSA is located in RAM for a system that has both RAM and ROM.

FSA is the area of memory above the VSA. The PL/M identifier MTMORY may be used to reference the FSA.

6.3.3 INTERP/80 for the 8080 Microprocessor

An INTERP/80 program available on the MITRE TSO system provides a software simulation of the Intel 8080 CPU, along with "free form" execution monitoring commands to aid in system software development for 8080-based microcomputers.

INTERP/80 accepts machine code produced by the PL/M 8080 cross-compiler, along with execution commands from a time-sharing terminal. The execution commands allow manipulation of the simulated system memory and the 8080 CPU registers. In addition, operation and instruction break-points may be set to stop execution at crucial points in the program. Tracing features are available which allow the CPU operation to be monitored. INTERP/80 also provides symbolic reference to memory storage locations as well as numeric reference in various number bases. Some of the commands available in INTERP/80 are presented in table 6-3. For a complete description of INTERP/80, see reference 5.

6.4 TEKTRONIX 8002A SOFTWARE DEVELOPMENT SYSTEM

The heart of the Tektronix software development system is the Tektronix 8002A microprocessor laboratory. The system architecture of the Tektronix 8002A centers around three microprocessors incorporated into circuit card modules. The system processor, using TEKDOS operating system commands, controls system activity such as organizing, storing, and retrieving system and user programs on the disc drives, executing the text and debug programs, supervising the emulator processor through separate debug hardware, and directing all I/O activity for various system peripherals such as the flexible disc unit, the system terminal, and a line printer. See figure 6-6 which shows the 8002A microprocessor laboratory used in conjunction with the Intel SBC 80/20 single board computer.

Table 6-3

INTERP/80 Commands

Command	<u>Function</u>		
LOAD	Causes symbol tables and code to be loded into the simulated MCS-80 memory.		
GO	Starts execution of the loaded 8080 code.		
(NO) INTER	Simulates an 8080 interrupt.		
TIME	Sets and displays the simulated 8080 cycle counter.		
CYCLE	Allows the simulated CPU to be stopped after a given number of cycles.		
(NO) TRACE	Enables tracing feature when particular portions of the program are executed.		
(NO) REFER	Causes the CPU simulation to stop when a particular storage location is referenced.		
(NO) ALTER	Causes the CPU simulation to stop when the contents of a particular memory location are altered.		
CONV	Displays the values of numbers coverted to the various number bases.		
DISPLAY	Displays memory locations, CPU registers, symbolic locations, and I/O ports.		
SET	Allows the values of memory locations, CPU registers, and I/O ports to be altered.		
BASE	Allows the default number base used for output to be changed.		
(NO) INPUT	Controls simulated 8080 input ports.		
(NO) OUTPUT	Controls simulated 8080 output ports.		
PUNCH	Causes output of machine code in BPNF or hexadecimal format.		
END	Terminates execution of an 8080 program.		



Figure 6-6 Tektronix 8002A Microprocessor Laboratory Workstation

The emulator processor, a system option for each microprocessor the 8002A can support, is controlled by the system processor. the debug module hardware, the emulator processor performs two functions. In emulation mode 0, this processor executes the user program so that run-time and logic errors can be detected before actual software/hardware integration begins. In emulation modes 1 and 2, the emulator processor, operating in conjunction with the prototype control probe, provides the capabilities for complete software/hardware debugging and integration. The control probe plugs into the microprocessor socket on the prototype circuit board, thereby enabling the emulator processor to take the place of the actual microprocessor that ultimately is installed in the user hardware. In emulation mode 1, the emulator processor executes the user program under supervision of the debugging system in program memory and memory mapped to the prototype. I/O and clock signals are also available to the user hardware. In emulation mode 2, the emulator processor executes the user program fully in the prototype hardware with all memory, I/O, and clocking functions made available. The debug system, however, still maintains control of the emulator processor.

The third microprocessor, the assembler, runs the relocatable macro assembly program when the TEKDOS assembly command is invoked. Upon completion of assembly, assembled object code is stored on disc memory in hexadecimal format. The assembler performs its function for each separate microprocessor supported by the 8002A with the installation of the appropriate assembler software.

Other circuit card modules within the 8002A provide supporting software and hardware functions. The system memory contains 16K bytes of dynamic RAM and a bootstrap loader resident in 256 bytes of ROM and is accessed only by the system processor. The system memory is automatically loaded with the resident portions of TEKDOS each time the 8002A system is powered up. It also provides buffer space for all I/O activities.

A maximum of 64K bytes of program memory resides on separate circuit-card modules within the 8002A mainframe and can be accessed by the system processor or the emulator processor. Program memory is used by the system processor as a text buffer during text editing sessions and is available during user program execution as a storage area in conjunction with the emulator processor. The feature of separate system/program memory structure allows the system memory to maintain its integrity at all times should the program crash for any reason, thus allowing the operating system software to remain intact.

System components within the 8002A are joined by a common system bus which is divided into a system side and a program side. The debug module provides the hardware interface between the system and emulator processors while the system communications module provides EIA standard RS-232C interfaces for all peripheral devices except the dual flexible disc drive unit and the associated system terminal.

Storage of about 660K bytes on dual disc drives (330K bytes per single-sided disc) is available as the mass storage medium for the 8002A. The drive unit communicates directly with the system processor module through an interconnecting cable.

The real-time trace module, an eight-clip test probe, and the data acquisition interface panel make up the real-time prototype analyzer of the 8002A. The module provides a high-speed buffer memory capable of retaining 128 data words in a dynamic fashion. Each acquired word, composed of 48 bits, contains 16 address bits, eight of 16 bits from the system bus, and eight data bits from the test probe. An additional eight bits identify the type of cycle such as read, write, I/O memory, or instruction fetch. The module also contains the controlling logic circuitry for utilizing the analyzer's command repertoire. The test probe itself is an eightpin, high-input impedance device which may be connected during emulation modes 1 and 2 to any locations on the user prototype hardware. Fixed TTL or variable threshold levels are switchselectable on the interface unit, with acquired data buffered by the probe and then fed to the real-time trace module via the cable interface. Data acquired by the trace module and the eight channels of externally acquired data are thus subject to the same TEKDOS command set of the 8002A. Finally, two BNC connectors on the data acquisition interface can also be used to trigger a logic analyzer or scope, thereby allowing greater trouble-shooting capability of prototype circuitry during program execution while maintaining debugging control through the 8002A.

The PROM programmer within the 8002A can support two PROM programming options, one for 1702A PROMs and one for 2704/2708 PROMs. Each option consists of the appropriate plug-in circuit card and PROM programming software to supervise and control the transfer of user programs between program memory and the PROM chip. Interface to the PROM programmer is on the front panel of the 8002A system mainframe, via zero-insertion-force sockets.

さん あると

The ability of the 8002A to communicate with external host processors provides the flexibility of writing software externally to the 8002A and downloading either source program or source code.

This is particularly useful when high-level language support is separately available, as is the case on the MITRE TSO system.

In summary, the 8002A is a complete design tool for software development to eventual software/hardware integration. Its powerful operating system software, TEKDOS, performs all utility routines, maintains the debugging system, provides emulation support routines, and controls the PROM programmer. The 8002A can support a variety of currently available 8-bit and 16-bit microprocessors and offers full prototype test and emulation capability at three progressive levels.

SECTION 7

CONCLUSIONS/RECOMMENDATIONS

The results of the improved polling project described lead to the following conclusions and recommendations:

- Microcomputer interfacing techniques offer a powerful technique for simulating new AFSATCOM satellite terminal polling performance capabilities.
- This successful microcomputer application was attributable to availability and acquisition (where necessary) of comprehensive computer hardware facilities and high-level language software tools designed to support personnel having little or no background in microprocessor-based techniques. Without these facilities and tools several additional personnel, particularly experienced programmers, would have been required.
- Polling improvements using existing AFSATCOM terminal interfaces via MPU software-only changes appear to offer significant new enhancements to a large segment of the AFSATCOM user community.
- Providing an on-line access to demonstrate these potential capabilities to AFSATCOM users without the need to modify either AFSATCOM hardware or software has proven to be cost-effective. This concept should be pursued in accordance with the follow-on demonstration objectives/plans described below.
- The microcomputer concept and approach described appears to also lend itself to other similar applications. The fully established facilities and proven software support tools for AFSATCOM at MITRE-Bedford should be used in the assessment of future terminal upgrades.
- The feasibility of serially adding microcomputer hardware to an existing AFSATCOM terminal for achieving growth capability has been demonstrated. With further development and suitable qualification of this additional hardware, modifying existing MPU software may be unnecessary.

The microcomputer system as presently configured is capable of operator interface via the RS-232 serial I/O port of its internal SBC 80/10 computer. Using this port, a Texas Instruments Model 765 Intelligent Terminal and a Lear Siegler ADM-3A cathode ray tube

(CRT) terminal were successfully interfaced, both locally and over telephone landlines, using a Bell 103J modem. Thus, a remotable I/O capability has been provided with commercial off-the-shelf hardware. The use of such a remoting capability makes future demonstration testing of polling improvements possible from any location having access to a telephone landline circuit and avoids the need to physically relocate the breadboard microcomputer from its interface at MITRE-Bedford to its collocated AFSATCOM Type 12 CP terminal.

Specific follow-on demonstration and evaluation objectives are derived from key issues related to incorporating polling improvements into the AFSATCOM System. Some of these objectives are:

- Evaluate the operational suitability of improved polling capabilities.
- Evaluate the enhancements resulting from improved polling in relation to performance of the AFSATCOM mission.
- Test and evaluate the effectiveness of the improved polling capabilities, including the operator-terminal interface, in passing message traffic between terminals netted in various polling modes.
- Test and evaluate the adequacy of the microcomputer hardware/software to support the overall objectives.
- Demonstrate the potential role of improved polling capabilities to all concerned agencies, using hands-on hardware/software.
- Develop a successful test-bed for simulating future AFSATCOM hardware/software polling improvements and for investigating other related areas.
- Obtain data and operating experience for an eventual upgrade of the AFSATCOM operations concept.

The improved polling test objectives cited will be fulfilled through three test categories. These are:

- Demonstration of functional interface capabilities
- Terminal performance tests
- Network tests

The second secon

No. of the Control of

The first category will show the basic interface capabilities and inherent limitations of the as-designed microcomputer hardware/software in the AFSATCOM environment. The second and third categories will quantify the effectiveness and suitability of the polling improvements at the terminal and network levels and obtain an estimate of actual polled network behavior in an operational environment. This follow-on demonstration can be conducted using the microcomputer hardware/software and the AFSATCOM Type 12 CP terminal located at MITRE-Bedford. This demonstration can be performed jointly by MITRE and Air Force personnel.

LIST OF REFERENCES

- Intel Systems Data Catalog 1980, Santa Clara, CA: Intel Corporation, August 1979.
- 2. SBC 80/20 and SBC 80/20-4 Single Board Computer Hardware Reference Manual, Santa Clara, CA: Intel Corporation.
- 3. 8008 and 8080 PL/M Programming Manual, Rev. A, Santa Clara, CA: Intel Corporation.
- 4. 8080 PL/M Compiler Operators Manual, Rev. A. Santa Clara, CA: Intel Corporation
- Intel INTERP/80 User's Manual, Rev A. Santa Clara, CA: Intel Corporation.

APPENDIX A

BASELINE DESCRIPTION OF THE AFSATCOM ROLL CALL POLLING MODE

All Air Force Satellite Communications System (AFSCS) terminals can provide two-way teletypewriter (TTY) record communications using frequency shift keying (FSK) modulation at a 75 bits per second (b/s) serial rate over standard ultra high frequency (UHF) channels. The Roll Call Polling mode is one of four modes possible on the narrow-band FSK channels; the other modes are Random, TDM-1, and TDM-2.

A.1 ROLL CALL POLLING SYSTEM

Polling may be conducted on either half-duplex or full-duplex circuits and requires both a Net Control Station (NCS) and network discipline. The NCS transmits Poll Call messages to each pollable automatic send/receive (ASR) unit in the network. Each ASR, when polled, is given a 30-second period of time in which to transmit a precomposed response to the NCS. When individual ASRs are in the poll mode, they are inhibited from transmitting unless they are polled.

Roll Call Polling is initiated by a command post (CP) terminal. Force terminals reply through use of a stored message buffer in the ASR containing the previously prepared response. Poll inquiry is then possible when the polling function is selected at the ASR device.

When the CP message-processor generated poll inquiry is detected at the polled terminal through use of a unique code sequence peculiar only to that particular terminal or ASR, the precomposed message is transmitted back to the CP terminal. Roll Call Polling is structured so that the CP NCS queries each terminal in the polling net in sequence and each polled terminal replies only upon detection of its own unique code.

The Roll Call message from the CP NCS is unclassified and structured as shown in table A-1 and table A-2.

Table A-l

Roll Call Message Format

Address Header

Address Header

Address Header

からまとなるとなるという

 ${\rm A_2}$ and ${\rm A_3}$ designate the particular group.

 \mathbf{A}_1 and \mathbf{A}_2 and \mathbf{A}_3 designate the particular member.

** The mode character found in the message address header is signified by the variable N representing address header codes.

Selective Inquiry message structure is closely related to Poll Call message structure and differs only by the change to the variable found in the mode character.

The address portion of Selective Inquiry messages is similar to the Poll Call message and consists of an SOH character denoting start-of-header and signifying that the next four characters are an address header. The first character following the SOH is the mode character whose codes are represented in table A-2. It should be noted that only when the mode character is 0 does it denote that the message is a Poll Call.

Table A-2
Address Header Codes

N = Mode	Function	
0	Poll Call	
1	Super Group Call	
2	Group Call	Selective Inquiry Modes
3	Individual Call	

^{*} A3 designates the particular super group.

An All Call mode, denoted by no address header, is accomplished by omission of the SOH designator, the mode character, and the three address designation characters.

The three address designator characters (second, third, and fourth characters following the SOH) use three hexadecimal digits (0-9, A-F) for A_1 , A_2 , and A_3 and result in 4096 possible addresses.

Each pollable device in the network must have a unique programmed address. This address is manually inserted into the ASR via thumbwheels and must be part of the Poll Call message. If there is a match, the precomposed message will be transmitted as a Poll Response message with an unclassified message structure, as follows:

UUU,
$$\overline{SOH}$$
, F, A₁, A₂, A₃, message (if any), \overline{ETX} ,

where F is the mode character indicating that the message is a Poll Response.

If there is no match, the Poll Response message is not transmitted.

A.2 CP TERMINAL POLLING

The message processor unit (MPU) within the CP terminal can accommodate four poll groups, with each group containing up to 16 members for a total of 64 members. Members of any one group are restricted to the same combination of second and third (A $_2$ and A $_3$) address characters. Two polling commands are defined. One command causes all members of a single group to be polled. The other command causes all members of all groups currently defined to be polled.

Upon reception of a polling communications supervisory (COMSUP) command from an ASR, the MPU transmits a poll request message sequentially to each member of the group, waiting 30 seconds between requests for replies. A poll response is timed out if not received in its entirety within 30 seconds. The MPU will automatically change the individual address character (A₁) designating a member of a group with each successive poll request message according to the stored poll table in the MPU. This poll table can be either entered or changed by COMSUP commands, which provide for the dynamic definition and maintenance of up to four concurrent poll groups. New groups can be added, existing groups deleted, or individual members added or deleted from existing groups. See table A-3 for a listing of COMSUP commands applicable to polling.

Table A-3
COMSUP Commands Used in Polling Operations

COMSUP Command Description 1) $\overline{P}ATG gg \left[11 \mid a \mid .. \right] \overline{c}$ ADD TO/CREATE POLL GROUP 2-character group identifier where gg = { all alphabetic characters are upper case. Characters for "gg" must be selected from the hexadecimal group \emptyset through 9, A through F. Optional 2-character line mnemonic 11 = to poll on; the line mnemonic is specified only when defining a new (Aircraft identifier (one to eight characters may be specified in one a = { command). All alphabetic characters are upper case. Characters "a" must be selected from the hexadecimal group Ø through 9, A through F. 2) $\overline{P}DEI. gg \left\{ a \right\} .. \overline{c}$ DELETE FROM POLL GROUP 2-character group identifier where gg = { all alphabetic characters are upper Aircraft identifier (one to eight characters may be specified). If none is specified, the entire group a = will be deleted. All alphabetic characters are upper case. Characters for "a" must be selected from the hexadecimal group Ø through 9, A through 9. 3) PPLG gg c POLL GROUP $gg = \begin{cases} 2-character group identifier where \\ all alphabetic characters are upper \end{cases}$ case. 4) \overline{PPAG} \overline{C} POLL ALL GROUPS

Table A-3 (Continued)

Notes

The group tables are built and maintained dynamically in response to operator commands. A new group will be defined as a result of an ATG command that calls out a group not already in the tables. Members in a group are identified in the ATG and DEL commands by three characters; the last two characters of a group member identifier are the same as the group identifier. When all group tables are in use, the form of the command used for defining up to four groups at any given time will be rejected by the MPU. Another form of the command is available for use in subsequently adding additional members to an existing group. Attempts to add a member already in the group will cause the entire command to be rejected.

For the command used to delete individual members from a polling group, the entire command is processed and all identifiers are validated by the MPU before the group table entry is modified. Any error detected will cause the entire group to be rejected. Another form of this deletion command is available for use in deleting an entire group and releasing the associated table entry. It should be noted that the first form of this command can also result in the group table entry being released if all its members are released.

Each member is polled by building a Poll Call message containing the group and member identifiers, then queuing the message to the MPU output line associated with the group.

When the Poll Response message is received, it is verified and forwarded to a high speed printer with the prefix POLL RESPONSE RX. The next group is then polled and the process continues until the entire group has been polled. If no response is received from a particular member within 30 seconds, a print-out alarm is generated to indentify the group member who failed to respond. A received response from a terminal not polled (e.g., address of responding terminal does not match that in the poll request message) results in the message being intercepted, forwarded for printout on the high speed printer, and a POLL RESPONSE ER interrupt message being printed out along with the Poll Response. See table A-4 for a description of the MPU Status/Alarm and Intercept Messages for polling.

Table A-4
MPU Status/Alarm and Intercept Messages

	Condition	Status/Alarm Message
1)	GROUP XX POLL COMPLETE XX 2 hexadecimal digits, a group identifier.	PLG XX COM
2)	ALL CROUP POLL COMPLETE Processor has completed polling all previously defined groups.	PAG COM
3)	NO RESPONSE TO POLL A time-out has occurred on a particular member.	POLL RSP T/O
	Reason for Intercept	Intercept Message
4)	Poll Response from station not polled	POLL RESPONSE EP
5)	Poll Response received	POLL RESPONSE RX
Not	e•	

Note:

MPU-generated Status/Alarm messages are prefixed with a datetime header and a line identification mnemonic which indicates the source of the message.

A.3 FORCE TERMINAL POLLING

As a member of a poll, the Force terminal pollee must dial his proper address into the ASR thumbwheels, enable the address recognition switch on the ASR, and press the poll XMT button once. Whenever the ASR receives a message header of the format following the UUU consisting of:

$$\underline{SOH}$$
, β , A_1 , A_2 , A_3

where

$$A_1$$
, A_2 , A_3 is the ASR address,

the ASR will automatically transmit whatever is in ASR storage up to the message length selected, each time the ASR is polled. If a change to the Poll Response message is desired, the operator must press the compose and edit button and make the changes, thereby taking the particular ASR out of the automatic poll response mode. Thus, poll responses are inhibited while the Poll Response messages are either updated or revised. After update, the poll XMT button must be depressed again to return to the poll. Both Poll Call and Poll Response messages should print out on the pollee's ASR. In addition, any text attached to the Poll Call message will not be printed out by the pollee's ASR, regardless of model ASR (i.e., 120A; 120B; 129; Mini-I/O). In actual operation during an automatic poll request, the model 120A ASR prints nothing; models 120B, 129, and the Mini-I/O print only the poll message (five characters described above) along with the ETX (whenever it occurs).

A.4 OTHER MPU-EQUIPPED TERMINAL POLLING

An MPU-equipped terminal can also be polled as described above but is limited to a shorter Poll Response message because of the message transfer technique inherent in the MPU-equipped terminal. Since data are transferred from the modem to ASR on a message-by-message basis, the complete Poll Response message is transferred into the processor from the ASR before going to the modem. To stay within the CP processor poll-out time of 30 seconds, the maximum length of a Poll Response message from a processor equipped terminal is 130 characters.

APPENDIX B

PL/M-80 SOURCE CODE

```
III.DATA - MCL 2: VERSION 1 - 10 JULY 80 hhH.DATA - MCL 1: VERSION 3 - 10 JAN. 80
                                                                                  00000500
/*
                                                                                  00001000
1001
                                                                                  00002000
/•
            IMPROVED FOLLING COMSUF COMMANDS
                                                                                  00003000
10
       INADLE CHECKIN MODE
                                                    ZNR!SABC TXXX
                                                                                  00004000
      LISABLE CHECKIN MODE
                                                    ZNR!SABCOXXX
                                                                     */
                                                                                  00005000
ICI TABLE CHECKIN
                                                                     */
                                                                                  00006000
                                                    2NR!CABCAAA1
                                                                     */
      KCI TABLE CHECKIN
                                                                                  00007000
                                                    2 NR! DABCBBB 1
      NCI TABLE CHECKIN
                                                    ZNR!EABCCCCO
                                                                                  00008000
                                                    ZNR! BABCDD1
      UCI TABLE CHECKIN
                                                                                  00009000
      PRICEITY TABLE LOAD
                                                    ANR! TABC....
                                                                                  00010000
                                                                     */
      BULTINE TAFLE LOAD
                                                    ZNR!ZABC....
                                                                                  00011000
      NU-THAPPIC TABLE LOAD
                                                    ZNR! BAEC....
                                                                                  00612000
      LISABLE CONSUP MSG PRINTOUT
                                                    ZNR14ADC0
ZNR14ABC1
                                                                                  00613000
      ENABLE COMSUP MSG PRINTOLT
                                                                                  JUC 14000
      FE-INITIALIZE FCI TABLE
                                                    ZNR!FABC
                                                                                  J0615000
       RE-INITIALIZE HOL TABLE
                                                    ZNR!GAbC
                                                                                  00016000
      RE-INITIALIZE NCI TABLE
                                                                                 0001700u
                                                    ZNR!HABC
      EE-INITIALIZE GCI TABLE
                                                    ZNR! AAFC
                                                                                  JU016000
      PRICE THE CHECKIN TABLE TRANSPER FOLTING CHECKIN TABLE TRANSPER
                                                    ZNR!TABCXXYZ
                                                                                 00619666
                                                                     •/
                                                    ZAH! TAECYXYZ
                                                                                  00020000
      NO-TENEFIC CHECKIN TABLE THANSFER
                                                    ZNH!TAbCZXYZ
                                                                                  00021000
                                                                     */
      FULL TABLE THANSFER
                                                    2 AR! TAPCTXYZ
                                                                                  00022000
11111111
      ibili o Phinicut Poll Table
                                                                                 00023000
                                                    ZAR!MABC
                                                                      */
      FULL TABLE FRINTOUT (LOCAL UNIT)
FULL TABLE FRINTOUT (REMOTE ALL-CALL)
                                                    ZNK!UAECZXYZL
                                                                                  00024000
                                                                                  00045000
                                                    2NR!UABCZXYZR
                                                                     */
      INIGHTY TABLE PHINTOUT NOUT TABLE FRINTOUT
                                                    2 NH! UABCPXY2
                                                                                  00026000
                                                    ZNK!UAFCRXYZ
                                                                     */
                                                                                  00027000
      NO-TEAFFIC TABLE FRINTOUT
                                                    ZNR!UABCQXYZ
                                                                                  00028000
      FOLE PARTE (SINGLE POLL)
      GROUP TABLE PRINTOUT
                                                    2NH!UABCOXYZ
                                                                                  00029000
                                                    ZNR! JABC 15
                                                                                  00030000
      FOIL ENABLE (MULTIPLE POLL)
                                                    ZNR!JABC1M
                                                                                  00031000
                                                    ZNR!JAECOS
      FULL LISABLE (SINGLE)
                                                                                  00032000
      FOLL INTERPUTT ENABLE (NON-TUR)
                                                    ZNR!LABC10
                                                                                  00033000
      FOLL INTERRUFT ENABLE (TER & NON-TOR)
FOLL INTERRUFT DISABLE (TER & NON-TOR)
                                                    ZNH!LABCO1
                                                    ZNR!LABC11
                                                                                  00035000
/•
                                                    ZNR!LABCOO
                                                                                 00036000
      SHIUFITY CHECKIN TABLE DELETION
                                                    ZNR! NAECPOI
                                                                                 00037000
       BOUTING CHECKIN TABLE PELETION
                                                                                  30036000
                                                    ZNR!NABCRU1
       NO-THANFIC CHECKIN TABLE DELETION
                                                    2NH!NAECNO1
                                                                                  00039000
      GNOUS CHECKIN TABLE DELETION
                                                                                 30040000
                                                    2NR!NAbCG01
/* FFFCHTHACK & FOLL SLOT REASSIGNMENT
                                                    ZNH!OABC ....
                                                                                  00041000
     ELAC ENAFLE (NCN-TDM)
                                                    ZNR!PABC1
                                                                                  06642000
      LUAC LISABLE (NON-TOM)
                                                    ZNH!PABCO
                                                                                  00043000
10
      MESSAGE TRANSMISSION DELAY (NON-TOM)
                                                    ZAK! KAHC 10
                                                                                  00044000
10
      ENABLE ALL TYPES OF MESSAGES
                                                    ZNR!VALCO
                                                                                  00045000
10
      LICHTGARD NON-COMSUP MSGS while POLLING ZARIVABOT
                                                                                  00046000
      DISHEGARE ALL NON-COMSUP MISSAGES
                                                    ZNR!VABC2
      FACGAPM RESTART (RE-IFT MICHUFROCESSUR) ZAR! RABC
RS.3. INTERFACE ENABLE(1)/LISABLE(0) ZAR! QABC1
                                                    ZNR!QABC1
10
1.
      MICHCLECESSCH ACDRESS CHANGE
                                                    ZNR! IABCCAB
                                                                                  00050000
/**/
                                                                                  00051000
100%
                                                                                  00052000
                                                                                  00053000
    1) ALL CUTSUF RSGS REQUIRE "LNK!" FOLIOHEL BY AT LEAST ONE MULE CHARACTER IMMEDIATELY AFTER THE "!".
                                                                                  00054000
                                                                        •/
                                                                                  00055000
    2) TWO ETX'S ARE PART OF THE COMSUP MSG SUFFIX. */
3) "XYZ" IS INLICATIVE OF A HEXIDECIMAL DESTINATION ADDRESS */
*) "ABO" IS INDICATIVE OF THE INITIALIZED ADDRESS OF THE */
                                                                                  00056000
                                                                                  00057000
                                                                                  00056000
/*
        MICACPROCESSON; THIS IS A COMSUP VARIABLE IN ITSELF.
                                                                                  00059000
144/
                                                                                  00060000
1 * 4/
                                                                                  00061000
    INPUT-/ CUTFUT DESCRIPTIONS
```

F

```
INPUT 1 = CHAR READY FROM MODEN
                                                                                      00063000
              INPUT 101 = CHAR READY FROM MPU
                                                                                      00064000
              INPUT 2 = CONTINUE PROGRAM
                                                                                      00065000
              00066000
                                                                                      00067000
                                                                                      00070000
                                                                                      00071000
                                                                                      00072000
                                                                                      00073000
                                                                                      00074000
                                                                                      00075000
                                                                                      00076000
                       6253 PROGRAMMABLE COUNTER INPUT/OUTPUT
                                                                                      00077000
        OUTFUT 22G (DCH) = LOAD/READ COUNTER NO. 0 (8253)
OUTFUT 222 (DEH) = LOAD/READ COUNTER NO. 2 (8253)
OUTPUT 223 (DFH) = WRITE NODE WORL/NO-OP (3-STATE)
                                                                                      00077010
                                                                                      00078000
                                                                                      00079000
                                                                                      00060000
              OUTPUT 1 - DATA TO EPU
                                                                                      00061000
              OUTPUT 101 = DATA TO MULEN
                                                                                      00062000
            CUTFUT 2 = DIAGNOSTICS
                                                                                      00063000
              OUTFUT <4>: 233 (E9H)
                                        - PARITY DISABLE TO HODEM
                                                                                      00084060
                                           AND I/O CHIF RESET
                                                                                      000E5000
              OUTPUT <6>: 234 (EAH) = RX ENABLE TO HPU
CUTFUT <106>: 235 (EEH) = I/O TX ENABLE TO HODEN
                                                                                      00066000
                                                                                    */00087000
                                       AND RX CLOCK ENABLE TO HPU
                                                                                   ·/ 00068000
                                                                                      00089000
                     hs232 INPUT/OUTILT
                                                                                   */ 00040000
              OUTPUT 237 (EDH) = CCNTROL WORD (CPU TO USART)
OUTPUT 236 (ECH) = LATA (CPU TO USART)
INPUT 237 (EDH) = STATUS WORD (USART TO CPU)
INPUT 236 (ECH) = DATA (USART TO CPU)
                                                                                   •/ 00091000
                                                                                   •/ 00092000
                                                                                   •/ 00093000
                                                                                   . 00094000
                                                                                      06095000
                                                                                      00096000
                                                                                      00097000
                                                                                      00096000
              MEMORY ALDRESSES FOR 1/C
                                                        +/
                                                                                      00099000
                   INFUTILADE - OFFCOR
                                                                                      00 100 000
                   INPUTSTADE = OFF4CH
                                                                                      96 10 1000
                   CUTFUTISAUR . OFE4CH
                                                                                      90 10 2000
                   INFUT 10 1SADE = OFECCH
                                                                                      00 10 3000
                   INFUT1035ADR - CFE4CH
                                                                                      00104000
                   INPUT 105SADE - CFFUCH
                                                                                      00105000
                   OUTPUTICISADE = OFF40H
                                                                                      00106000
                   INPUTSSADE - OFECOH
                                                                                      00 10 7000
                                                                                      00 10 8000
                                                                                      00 10 9 0 0 0
                                                                                      30110000
100/
                                                                                      00111000
LECLARE (PROGRAMSSTAFT, INITIALINSG) LABEL;
                                                                                      00 112000
/* kS232 1/0 CHIP INITIALIZATION */
                                                                                      00113000
/* MODE CONTROL */
PROGRAMSSTART: OUTPUT (237) = 79h;
                                                                                      00 114000
                                                                                      00115000
/+ 79h = C111 1CC18; C1=1 STOP BIT/1=EVEN PARITY/1=PARITY ENABLE +/
                                                                                      00 1150 10
                          10=7 BIT CHARACTER/01=1x BAUD RATE FACTOR */
                                                                                      00115020
            / COMMAND INSTRUCTION - FESET */
                                                                                      00116000
            CUTPUT (237) = 40H;
                                                                                      00117000
            /* MODE INSTRUCTION */
                                                                                      00118000
            CUTFUT (237) = 79%;
                                                                                      00119000
            /* COMMAND INSTRUCTION - INITIALIZATION */
                                                                                      00120000
      CUTPUT (237) = 17H;
17H = 6001 0111E; 0=ENTER HUNI HODE/O=INTERNAL RESET/
                                                                                      00 12 1000
                                                                                      00 12 10 10
10
            O=RTS/1=ERROR RESET/O=SEND BREAK CHAR/1=RX ENABLE
                                                                                      00 12 1020
             1=DTR (NOT) OUTPUT FORCED TO ZERO/1=TX ENABLE
                                                                                      00 12 10 30
```

```
00 122000
     CONTROL WORD INITIALIZATION (8255 # 1 CONTROL ONLY)
                                                                                      00123000
            CUTPUT (231) = 9BH;
CUTFUT (235) = 90H;
                                                                                      00124000
                                                                                      00125000
            CUTPUT (233) = 10H;
                                                                                      00126000
            OUTPUT (233) = OH;
                                                                                      00127000
            CUTPUT (233) = 10H;
                                                                                      00128000
/**/
/* CUNTROL REGISTER INITIALIZATION
                                                                                      00129000
                                                                                      00130000
DECLARE MPUBLOSCILISPOINTER ADDRESS INITIAL (OFFOOH);
                                                                                      00131000
DECLARE MPUSIOSCIL 1 BASED MPUSIOSCIL 18 POINTER BYTE:
                                                                                      00132000
            HPUSIOSCTL1SPOINTER = OFFCOH;
                                                                                      00133000
            MPUSIOSCTLY = 03H NOR OFFH;
                                                                                      00134000
144/
                                                                                      00135000
DECLARE MPUSIOSCIL2SFOINTER ADDRESS INITIAL (GFF40H);
                                                                                      00136000
DECLARE RPUSIOSCTL2 BASED MPUSIOSCTL2SPOINTER BYTE;
                                                                                      00137000
           MIUSIOSCTL2SPOINTER = OFF40H;
                                                                                      00138000
                                                                                      00139000
            MEUSIOSCILL = 2FH XOR OFFH:
/**/
                                                                                      00140000
            MEUSIOSCILT = 43H XOR CFFh;
                                                                                      00 14 1000
            HPU$IO$CTL2 = 03H XOR OFFH;
                                                                                      00142000
            MPUSIOSCTL1 = OCON XOR OFFH;
                                                                                      00 14 3000
100/
                                                                                      00144000
144/
                                                                                       00 145000
DECLARE MOLEMPIOSCILISPOINTER ALDRESS INITIAL (OFEOOH);
                                                                                      00146000
LECLARE HODERS 10 SCTL 1 BASED HODERS 1 CSCTL 1 SPOINTER BYTE;
                                                                                      00147600
            MCDEMSICSCILISPOINTER = OFEOOH;
                                                                                      00148000
            MCDEMSIGSCILT = 03H XOK OFFH;
                                                                                       00149000
100/
                                                                                       00 15 0 0 0 0
CECLARE MOLEMPIOSCILZSPOINTER ADDRESS INITIAL (CFE40H);
                                                                                       00 15 1000
LECLARE MOLEMETOSCILE BASED MODERFICSCILESPOINTER BYTE;

MCCEMSIOSCILESPOINTER = CFEMON;
                                                                                       00 15 2000
                                                                                       30 15 3000
            MCDEMSIGECTL2 = 2FH XGR OFFH:
                                                                                       00154000
1001
                                                                                       00155000
            MCDEMFICSCIL1 = 43H XOK CFFH;
                                                                                       30 156 000
            MUDINITOICTL2 = 03H XON CIPH;
                                                                                      00 15 7000
            HCBEMSICSCILT = OCOH XCR CFFH;
                                                                                       00158000
1001
                                                                                      00 15 9000
100/
                                                                                       00 16 0 000
/* INTRUCTION FOR DIVIDE BY 1240 (300 HZ) ON 16-BIT COUNTER # 2 */
                                                                                      00 16 1000
            CUIFUT (223) = 086H;
                                                                                       00162000
            CUIFUT (222) = ODBH;
CUIFUT (222) = O4H;
                                                                                       00 16 3000
                                                                                       00 164000
   INSTRUCTION FOR LIVIDE BY 4962 (75 HZ) ON COUNTER # 0
                                                                                       00 1640 10
    CUTPUT (223) = 36h;

CUTPUT (220) = 62h;

CUTPUT (220) = 13h;
                                                                                       00164020
                                                                                       00164030
                                                                                       00164040
/* These give 75 & 300 hz clocks fach 6.70 HHz On-BOARD XTAL */
                                                                                       00165000
1 * * /
                                                                                       00166000
/**/
                                                                                       00167000
DECLARE RS232$FLAG BYTE INITIAL (C);
                                                                                       00168000
             ES2328FLAG = 0;
                                                                                       00169000
DECLARE (RS232$SPACESTEST, RS232$OUTCHAR, RS232$INPUT1,
                                                                                      00 170000
             kS232$GC1, RS232$HODE$SET) LABEL;
                                                                                       00 17 1000
DECLARE (ALM), ADR2, ADR3 ) BYTE INITIAL ('A', 'B', 'C');
                                                                                       00 17 2000
            ALR1 = 'A';
                                                                                       00173000
            ADR2 = "B";
                                                                                       00174000
            ALA3 - 'C';
                                                                                       00175000
DECLARE (FOLLSSLOT1H, POLLSSLOT2H, FCLLSSLOT3H, POLLSSLOT4H, POLLSSLOT5H, POLLSSLOT6H, POLLSSLOT1L, POLLSSLOT2L, POLLSSLOT3L,
                                                                                      00176000
                                                                                      00177000
   FOLLSSLOTAL, POLLSSLOTSL, POLLSSLOTGL, TXSSLOTSINITIALH, 00178000
TXSSLCTSINITIALL, TURSPOLLSTXSSLCTSINITIAL) BYTE 00179000
INITIAL('1','1','2','3','4','5','1','9','7','5','3','1','0','1',1);00160000
```

The state of the section of the sect

```
FOLL#SLOT1H = '1';
                                                                                   00 18 1000
         PO113510T11 = '1';
                                                                                    00182000
         FOLLSSLOT2H = '1';
                                                                                    00 16 3 0 0 0
         POIL$SLOT21 = '9';
                                                                                   00 184000
         FOILSSIOT3H = '2':
                                                                                   00 18 5000
         1011$510131 = '7';
                                                                                    00 186 000
         FOLLESLOT4H = '3';
                                                                                    00 18 70 00
         i0113510T41 = '5';
                                                                                   00188000
         FOILSSLOTSH = '4';
                                                                                    00189000
         FOLLSSLOTSL = '3';
                                                                                    00 190 000
         1011451076H = '5';
                                                                                    00 19 1000
         PO115510761 = '1';
                                                                                   00192000
         TAISLOTSINITIALH = '0';
                                                                                    00 19 3000
         TXISLOTSINITIALL = '1';
                                                                                    00194000
         IDMSPOLLSTXSSLOTSINITIAL = 4;
                                                                                    00 195000
LECLARE (CS3, CS4) BYTE;
                                                                                    00196000
DECLARE (PC1LEXHIT, FOLISHODESERROR, POLLSOVERSTEST, XX, YY, WW, ZZ,
                                                                                    00197000
      REPOLLSTEST, POLLSMSGSCOMPLETE, OUTTABLE 18 STATUS STEST,
                                                                                    00 198000
      PULLINGSHODESTEST, AUTOSPOLLSHESTART, NONSTONSPOLL, EDACSHOEESSET, 00 199000
      TUMSPOLL, PCILSTABLESTY, POLLSINTERRUPT, RETRANSMITSFLAGSSET) 1ABEL: 00200000
DECLARE FOLISMSGSCOUNT ADDRESS INITIAL (0):
                                                                                    00201000
           FULLSMSG$COUNT = 0;
                                                                                    00202000
DECLARE ATLESEDACSTABLE (7) BYTE:
                                                                                    00203000
DECLARE (ECACEFIAG, REPOLLSFLAGSCOURT) BYTE INITIAL (0,0);
LDACSFLAG = 0;
                                                                                    00204000
                                                                                    00205000
           FEPULLSFLAGSCOUNT = 0;
                                                                                    00206000
DECLARE (CCEPRESENT, CCISPRESENT) ADDRESS INITIAL (0,0);
                                                                                    00207000
         CCAFRESENT = 0:
                                                                                    00208000
         CC 12 FALSENT = 0;
                                                                                    00209000
LECLARE ( IFOLI 101, GC1, OPOLL 101, IPOLL 1011, NONSTDM & DELAY 1, POLL SMODE, 00210000 DELAY 1 FOUR THE LATSHSG SSET, MCDEM SOUTPUT SCOMPOSE 1, 00211000
LCT1, GUIFLAG1, ECHTEST1, SPACETEST1, NOBSTDRSDELATSSET, CHAR4T1, 00212000 COMSUFLGAD1, GUICHAK1, NOCOMSUPLOAL1, KX$BUSY$OVERRIDE1, IPOLL11) 1ABE1;00213000
LECLARE ( SHIFTDOWN, SHIFTDOWN), LCADADR, ENDLOAD, ADRERROR 1 ) LABEL ; 00214000
DECLARE EATENSUSFLAGI BYTE INITIAL (CIH);
     EATERSUSPLACE - OIM:
                                                                                    00216000
LECLARE (CLET, CIRF), DELAY), CIR2, CIRP2, CIR3,
CIRF3) EYIE INITIAL (0,0,0,0,0,0);
LECLARE (CIRCNT), CIRCRT2, CIRCRT2) ADDRESS INITIAL (0,0,0);
                                                                                    00217000
                                                                                    00218000
                                                                                    00219000
           C1K1 = 0:
                                                                                    00220000
           CLAF1 = 6;
                                                                                    00221000
           CINCAT1 = U:
                                                                                    00222000
           LELAY1 . U:
                                                                                    00223000
           CLK2 = 0;
                                                                                    00224000
            CLKE2 = 0;
                                                                                    00225000
           LIKCHT2 = L;
                                                                                    00226000
           C1K3 = 0;
                                                                                    00227000
           CLKE3 = 0;
                                                                                    00228000
           LIKCHT3 = 6:
LECLARE (BUFFERSCELAY), NONSTDMSMSUSUELAY) ALDRESS INITIAL (48,300H); 00230000 EUFFERSDELAY) = 46;
           AGNSTDHSHSGSDELAY1 = 3UCH;
                                                                                    00232000
LECLARE (ELACSPOUTINE, OUTPUTSCONFUSE, PTTSLOADI, DELAYSCAL,
                                                                                    00233000
   ATTELCALI, ATTSLOADI, TTELCALEINDI) LABEL;
                                                                                    00234000
00235000
                                                                                    00236000
           ICLNE = 0;
                                                                                    00237000
           ICLKCAT = 0;
                                                                                    00238000
DECLARE (MSGECVERSDELAY, TCLK1, 1CLFP1, TCLKCAT1) BYTE
                                                                                    00239000
         INITIAL (64, C, O, O);
PSG$GVER$LELAY = 60H;
                                                                                    00240000
                                                                                    00241000
           TOLK! # 6:
                                                                                    00242000
           TCLKP1 . Č;
                                                                                    00243000
```

٠

" Total Page !

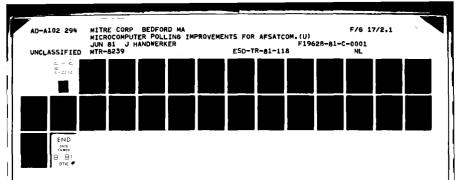
```
ICLKCHT1 = 0;
                                                                                   00244000
DECLARE (RETRANSMITSFLAG, TDMSPOLLSTX$SLOT) BYTE INITIAL (0, 1);
                                                                                   00245000
            HETRANSHITSFLAG = 0;
                                                                                   00246000
            TDM$F011$TX$SLOT = 1;
                                                                                   00247000
DECLARE (SLOTH, SLOTL, SLOT, SLOTSCOURT) BYTE INITIAL (0,0,0,0);
                                                                                   00248000
            SLOTH = 0:
                                                                                   00249000
            SIOTL = 0:
                                                                                   00250000
            SLOT = 0;
                                                                                   00251000
            SIGTSCOUNT = 0:
                                                                                   00252000
DECLARE (NILHCELAYH, NIDMDELAYL) ALTRESS INITIAL(0,0);
NILHDELAYH = 0;
                                                                                   00253000
                                                                                   00254000
NILHDELAYL = 0;
DECLARE SLCT&GUARD$TIME ADDRESS INITIAL (20H);
                                                                                   00255000
                                                                                   00256000
          SICTEGUARDSTINE = 20H:
                                                                                   00257000
DECLARE (TLMSFLAG, POLLSFLAG, AUTOSKEPOLLSFLAG) BYTE INITIAL ('0', 0, 0);00258000
           TERSFLAG = '0';
                                                                                   00259000
          PCILIFIAG = 0:
                                                                                   00260000
           AUTOSREPOLISFIAG = 0;
                                                                                   00261000
DECLARE (POLLETABLESTXSFLAG, NTDMSFCLLSINTERRUPTSFLAG,
                                                                                   00262000
   TDHiPGLLSINTERRUPTSFLAG) BYTE INITIAL(0,0,0);
PCLLSTABLESTXSFLAG = 0;
                                                                                   00263000
                                                                                   00264000
           NIUMSPOLISINTERRUPTSFLAG = 0;
                                                                                   00265000
          TEMSFOLLSINTERRUPTSFLAG = C;
                                                                                   00266000
DECLARE ( CS1, Chahl, OUT1,
                                                                                   00267000
   WAITING?, SPACE?, LINECOUNT) BYTE:
                                                                                   00268000
          GUT1 = 0;
HAITING1 = 0;
                                                                                   00269000
                                                                                   00270000
DECLARE CHARSTORE 1 (2000) BYTE;
                                                                                   00271000
DECLARE MSGCOUNT1(150) ADDRESS;
                                                                                   00272000
LECLARE OUTTABLE 1 (2000) BYTE;
                                                                                   00273000
LECLARE ( 11, TC1, OUTCOUNT1,CC1,J) ADDRESS INITIAL (0,0,1,0,0);
                                                                                   00274000
                                                                                   00275000
          11 = C;
          T(1 = 0:
                                                                                   00 ∠ 7 7 0 0 0
          OUTCOUNT1 = 1;
                                                                                   00278000
          CC1 = 0;
DECLARE CSIDATASPOINTER ADDRESS INITIAL (OFFOOR);
                                                                                   00283000
DECLARE CSTLATA BASED CS$DATA$POINTER BYTE:
                                                                                   00 34000
          CSSDATASFGINTER = OFFOOH;
                                                                                   00285000
DECLARE INFUT3$CATA$FOINTER ADDRESS INITIAL (OFF40H);
                                                                                   00286000
DECLARE INFUT3SCATA EASED INPUT3$DATA$POINTER BYTE;
                                                                                   00257000
          INPUT3$DATA$POINTER = OFF4GH;
                                                                                   00268000
DECLARE INFUTSEDATASFOINTER ADDRESS INITIAL (OFEOOH);
DECLARE INFUTSSLATA BASED INPUTSIDATASPOINTER BYTE;
                                                                                   00289000
                                                                                   00290000
INFUTSSDATASFOINTER = OFEOOH;

DECLARE CUTFUT ISDATASFOINTER ADDRESS INITIAL (OFEOH);

LECLARE CUTFUT ISDATA BASED OUTPUT ISLATASPOINTER BYTE;

CUTFUT ISLATASPOINTER = OFEOCH;
                                                                                   00291000
                                                                                   CO292000
                                                                                   00293000
                                                                                   00294000
LECLARE CS 1 CATA POINTER ADDRESS INITIAL (OFEOOH);
                                                                                   00295000
DECLARE CS 15 DATA BASED CS 15 DATAS POINTER BYTE;
                                                                                   00296000
         CS 11DATASPOINTER - CFEOOH;
                                                                                   00.57000
DECLARE INFUT 103 SDATA SPOINTER ADLRESS INITIAL (OFE40H);
                                                                                   00298000
LECLARE INFUTIGISDATA BASED INPUTIO3SDATASPOINTER BYTE;
                                                                                   00259000
         INIUT 103$CATA$PGINTER = UF140H;
                                                                                   00360000
DECLARE INFUTIG5$CATA$POINTER ALLRESS INITIAL (OFFOOH);
                                                                                   00301000
LECLARE INSUT1058DATA BASED INPUT1058DATASPOINTER BYTE;
                                                                                   00362000
         INFUTICS SUATAS POINTER = OFFCOH;
                                                                                   00303000
DECLARE OUTFUT 10 1 CATASPOINTER ALCHESS INITIAL (OFF40h);
                                                                                   00304000
DECLARE CUTEUT 10 18 DATA BASED OUTPUT 10 18 DATAS POINTER BYTE;
                                                                                   00305000
         CUTEUT 10 18 DATASFOINTER - CFF40H;
                                                                                   00306000
100/
                                                                                   00367000
                                                                                   00308000
DECLARE (CS, CHAR, ENC, MJF, MODE,
                                                                                   00309000
```

P. A.



```
00310000
            WAITING, SPACE) BYTE;
             ENC = 0;
MJF = 0;
                                                                                             00311000
                                                                                             00312000
DECLARE (N. NUMSGCI, TOTALSHUMSCI, NUMSFCI, PCISPOINTER, NC, NUMSFCI, RCISPOINTER, GCISPOINTER, BUHSKCI, NCLISPOINTER, CLKCNT, TCINITIAL, TCINITIAL) ADDRESS;

TOTALSHUMSCI = 0;
                                                                                             00313000
                                                                                             00314000
                                                                                             00315000
                                                                                             00316000
                                                                                             00317000
             NUMSGCI = 0;
NUMSPCI = 0;
                                                                                             00318000
                                                                                             00319000
             NUMSRCI = C:
                                                                                             00320000
             NUMSHCI = C;
                                                                                             00321000
             CLKCAT = 0:
                                                                                             00322000
                                                                                             00323000
             TCINITIAL = 0;
             10 TINITIAL = 0;
                                                                                             00324000
DECLARE (CLK, CLKF, DELAY) BYTE INITIAL (0,0,0);
                                                                                             00325000
             CLK = 0;
                                                                                             00326000
             CLKF = 0;
                                                                                             00327000
             CELAY = C:
                                                                                             00326000
DECLARE (BUFFERSDELAY, MSGSDELAY) ADDRESS INITIAL (48, 640);
BUFFERSDELAY = 48;
                                                                                             00329000
                                                                                             00330000
                                                                                             00331000
             MSG$DELAY = 160;
DECLARE ( 110111, GC, OPOLL1, QUIT, ST, MJT1, LCT, CHAR3T, CHAK4T, CHAR5T, CS$TEST, PRB$1CAD, RRB$10AD, NRB$10AD, MJT2, FIT$10AD$END, KTT$10AD$END, NTT$10AD$END, C$MODESCANCEL
                                                                                             00332000
                                                                                             00333000
                                                                                             00334000
     PTISLOAL, KTISLOAD, NTISLOAD, OUTSTABLESBUILD, C$MODESCANCEL, PC151L51, CCSTEST, KCISTEST, KCISTEST, CISNUMSTEST, C$MODESCANCEL, LCMSTEST, SPACESTEST, DELAYSKCUTINE, OUTSCHAR) LABEL;
                                                                                             00335000
                                                                                             00336000
                                                                                             00337000
DECLARE (PCIRESET, RCIRESET, NCIRESET, GCIRESET, ADRCHANGE) LABEL;
                                                                                             00338000
DECLARE ( STARSINSERT, MPUSTABLESBUILD ) LABEL ;
                                                                                             00339000
DECLARE ( FCISTORE, RCISTORE, RCISTORE ) (365) BYTE;
                                                                                             00340000
DECLARE GCISTORE (321) BYTE;
                                                                                             00341000
DECLARE CHARSSTORE (2000) BYTE:
                                                                                             00342000
DECLARE MSGSCOUNT (150) ADDRESS:
                                                                                             00343000
   DO 11 = 0 TO 149;
                                                                                             00344000
        MSGCCUNT (I1) = 0;
                                                                                             00345000
                                                                                             00346000
   END;
                                                                                             00347000
LECLARE (ENCHYPT, ENCTEST ) LABEL;
DECLARE SON LITERALLY '01H';
                                                                                             00348000
DECLARE CRHODE BYTE INITIAL (OH) ;
                                                                                             00349000
             C$MODE = OH;
                                                                                             00350000
DECLARE FCISHEADER DATA ("*PRIORITY*CI:") ;
                                                                                             00351000
DECLARE (I,TC,OUTSCOUNT, CC) ADDRESS INITIAL (0,0,1,0);
                                                                                             00352000
             1 = 0;
                                                                                             00353000
             TC = 0:
                                                                                             00354000
             CUTSCOUNT = 1:
                                                                                             00355000
             CL = 0:
                                                                                             00356000
DECLARE SCISHEADER DATA ('SCI:') ;
                                                                                             00357000
DECLARE GUTSTABLE (2000) BYTE;
                                                                                             00358000
DECLARE ACISHEADER DATA ('NCI:');
DECLARE PRESHEADER DATA ('*FRIORITY*RB');
                                                                                             00359000
                                                                                             00360000
DECLARE HRESHEADER DATA ('RRE:') ;
                                                                                             00361000
CECLARE MRESHEADER CATA (*NRB: *)
                                                                                             00362000
DECLARE STISHEADER DATA ( ** PRIORITY *TT *);
                                                                                             00363000
DECLARE ATTEMEADER DATA ('RIT:') ;
                                                                                             00364000
DECLARE NTISHEADER DATA ('NTT:') ;
                                                                                             00365000
DECLARE ( COMPOSE), FT1, RT1, RT1, GT1, POLITABLEBUILD1, IPLSTEST, CHECKINGCLESET1, TTR1, TBLDUMF1, GTABLEBUILD1, MOLEERROR1, MPUCINUMTEST1, PTTR1, RTTR1, NTTR1,
                                                                                             00368000
                                                                                             00369000
  POLLITEI, PTABLEDURPI, RTABLEDURFI, RTABLEDURPI, GTABLEDURPI,
                                                                                             00370000
FOLITABLEBUNF 1, RTABLEBUILD 1, PTABLEBUILD 1, POLITABLE DUMP 2,
                                                                                             00371000
    ATABLEBUILDI, POLIDUMPBUILDI, GECLLBUILDI, CHECKINSDELETE,
                                                                                             00372000
```

THE REAL PROPERTY AND ADDRESS OF THE PARTY AND

```
MPOLIBUILD1, RPOLLBUILD1, OVERFLOW1, OVERFLOW2) LABEL;
                                                                             00373000
DECLARE FOLLTABLE (1473) BYTE;
                                                                             00374000
DECLARE (POLLPGINTER), PTICOUNT) ADDRESS INITIAL (0,0);
FOLLPGINTER1 = 0;
                                                                             00375000
                                                                             00376000
          PTICOUNT1 = 0;
                                                                             00377000
DECLARE ITCHEADER DATA
                                                                             00378000
   ('FRICRITY CHECKIN TABLE DUMP', ODH, OAH);
                                                                             00379000
DECLARE ATCHEADER DATA
                                                                             00360000
   ( 'HOUTINE CHECKIN TABLE DUMP', ODB, OAH);
                                                                             00381000
DECLARE NTCHEADER DATA
                                                                             00382000
   ('NO TRAFFIC CHECKIN TABLE DUMP', ODH, OAH);
                                                                             00383000
LECLARE GTEREADER DATA
                                                                             00384000
    ('GROUF CHECKIN TABLE DUMP', OCH, OAH );
                                                                             00385000
DECLASE ADSERBORNER DATA
                                                                             00386000
    ( *** INCORRECT AUR CHARACTERS ***);
                                                                             00387000
DECLARE ERRORHDE CATA

(*** INCORRECT HODE CHARACTER ***);
                                                                             00388000
                                                                             00389000
DECLARE IFRERORMER LATA
                                                                             00390000
   ('TABLE TRANSFER ERROR: NO CHECKINS');
                                                                             00391000
DECLARE OVERFLOWNER DATA
                                                                             00392000
   (*** CHECKINS AT THE LIMIT ***);
                                                                             00393000
DECLARE FOLLDUMPHER DATA
                                                                             00394000
    ('FOLL TABLE DUMF: CHANNEL A', ODH, OAH ) ;
                                                                             00395000
DECLARE FOLINSGOVER DATA
                                                                             00396000
    ( *** All FOLL MESSAGES LOADED ***);
                                                                             00397000
                                                                             00398000
144/
                                                                             00399000
1001
                                                                             00400000
144/
                                                                             00401000
/* INITIALIZATION MSG TO TERMINAL OPERATOR
                                                                             00402000
DECLARE STARTUPSHSG DATA
                                                                             00403000
  ('IMPROVEL FOLLING SOFTWARE - VERSION 3: 10 JAN. 1980 ');
                                                                             00404000
INITIALSHSG: WAITING - WAITING + 1;
                                                                             00405000
        io I = 1 TO 54;
                                                                             00406000
           CUTTABLE (TC + I) = STARTUPSMSG (I -1);
                                                                             00407000
                                                                             00408000
        ENL;
        10 = TC + 54;
        MSGCOUNT (WAITING) = 54;
                                                                             00410000
144/
                                                                             00411000
1001
                                                                             00412000
/**/
                                                                             00413000
/**/
                                                                             30414000
100/
                                                                             00415000
/**/
                                                                             00416000
144/
                                                                             00417000
1001
                                                                             00418000
/**/
                                                                             00419000
1001
                                                                             00420000
/*
          SLOT CALCULATION ALGORITHE
                                                                             00421000
IPOLL1: SLCT = INPUT(228);
                                                                             00422000
        SLUT . SLOT NOR OFFH;
                                                                             00423000
        SLCTL = SLCT AND OFOH;
                                                                             00424000
        SLCIL = SHk (SLOTL,4);
                                                                             00425000
        SLUTH = SLOT AND OFH;
                                                                             00426000
        SLOTSCOUNT = SLOTH * 10 + SLOTL;
IF SLOTSCOUNT <> 0 THEN TOMSFLAG = *1*;
                                                                             00427000
                                                                             00428000
           ELSE TORFFLAG = '0';
                                                                             00429000
        GO 10 POLISABIT;
                                                                             00430000
/* 5107 IS INFUTTED AS 2 BCD VALUES ON A SINGLE INPUT */
                                                                             00431000
10
    THE 2 BCD VALUES ARE MASKEL OUT INTO SLOTH AND SLOTE
                                                                             00432000
100/
                                                                             00433000
/**/
/*
                                                                             00434000
                                                                             00435000
           FOLLING ALGORITHM
```

```
00436000
100/
/**/
                                                                               00437000
POLLSANIT: IF POLLSFLAG <> OIN THEN GO TO IPOLLIN;
                                                                               00438000
            IF PT 1COUNT 1 < 4 THEN GO TO POLLSHODESE BROR;
                                                                               00439000
               ELSE GO TO POLLSOVERSTEST;
                                                                               00440000
POLLSMODESERROR: POLLSFLAG = 0;
                                                                               00441000
                                                                               00442000
                  GO TO MODEERROR 1;
FOLLSOVERSTEST: IF POLLSMSGSCOUNT >= TOTALSNUMSCI
                                                                               00443000
                    THEN GO TO REPOLISTEST;
                                                                               00444000
ELSE GO TO OUTTABLE1$STATUS$TEST;
REPULL$TEST: IF AUTOSREPOLL$FLAG = 01H
                                                                               00445000
                                                                               00446000
                                                                               00447000
                 THEN GO TO AUTOSPOLISRESTART;
              FOLLSFLAG = OH;
                                                                               00448000
                   GO TO POLISHSGSCCHPLETE:
                                                                               00449000
POLLSHSG&CCHPLETE: WAITING - WAITING + 1;
                                                                               00450000
            LO I = 1 TO 30;
                                                                               00451000
              OUTTABLE (TC +I) = POLLESGOVER (I-1);
                                                                               00452000
                                                                               00453000
            END:
            TC = TC + 30;
                                                                               00454000
            ESGCOUNT (FAITING) = 30;
                                                                               00455000
            FOLLSHSGSCOUNT = 0;
                                                                               00456000
            GO TO IPOLL 11;
                                                                               00457000
OUTTABLE ISSTATUS STEST: IF TC1 > 0 THEN GO TO 1POLL 11:
                                                                               00458000
ELSE GO TO POLLINGSHODESTEST; FOLLINGSHODESTEST: IF TDMSFLAG = '1' THEN GO TO TDMSPOLL;
                                                                               00459000
                                                                               00460000
                       ELSE GO TO NONSTORSPOLL;
                                                                               0046 1000
AUTOSPOLLS FESTART: POLLS MSG SCOUNT = 0;
TDMS POLLS TXS SLCT = TDMS POLLS TXS SLOTS INITIAL;
                                                                               00462000
                                                                               00463000
                       GO TO POLLS MSG & COMPLETE;
                                                                               00464000
                                                                               00465000
NORSTONSFOLL: IF NTDESPOLLSINTERRUPTSPLAG = OWN
                                                                               00466000
            THEN GO TO IPOLL 11;
IF RETRANSMITSFLAG <> 61H THEN
                                                                               00467000
                 POLLSHSGSCOUNT . PCLLSHSGSCOUNT + 1;
                                                                               00468000
              LISE RETRANSMITSFLAG = 0:
                                                                               00469000
             NON-TON POLL MSG COMFOSITION
                                                                               00470000
       WAITING! = WAITING! + 1;
                                                                               00471000
       OUTTABLET(TC1 + 1) = 01H;
                                                                               00472000
       OUTTABLE 1 (TC1 + 2) = '0';
                                                                               00473000
       DG 11 - 1 TO 3;
                                                                               00474000
          OUTTABLET (TC1 + 2 + I1) =
                                                                               00475000
               POLITABLET(IT + POLLERSGSCOUNT + 4 - 4);
                                                                               00476000
                                                                               00477000
        OUTTABLET (TC1 + 6) = 03H;
                                                                               00478000
       OUTTABLET (TC1 + 7) = 03H;
                                                                               00479000
                                                                               00460000
       ELAC TABLE LOAD
                                                                               00481000
   DO J * 1 TO 7;
                                                                               00482000
     ATDRIELACSTABLE(J) = OUTTABLE1(TC1 + J);
                                                                               00483000
   LAC:
                                                                               00484000
144/
                                                                               00485000
       IC1 = IC1 + 7;
                                                                               00486000
                                                                               00467000
       MSGCCUNT1 (WAITING1) = 7;
                                                                               00488000
       GO IC IPOLLIT;
TDMSFOLL: 15 TDMSFOLISINTERRUPTSFLAC = Olh
                                                                               00469000
                                                                               00490000
              THEN GO TO IPOLL11;
          IF SLOTSCOUNT - TOMSPOLISTESSLOT THEN
                                                                               00491000
              TURSPOLLSTASSIOT = TURSPOLLSTASSIOT + 1;
                                                                               00492000
           ELSE GO TO IPOLL 11;
                                                                               00493000
    IF TEMSPOLESTASSEOT >= TOMSPOLESTASSEOTSINITIAL + B THEM
                                                                               00494000
              TDMSPOLLSTXSSLOT = TDMSPOLLSTXSSLOTSINITIAL;
                                                                               00495000
    TEM FOLL MSG COMPOSITION */
IF TOTALSNUMSCI - POLLSMSGSCOUNT >= 6 THEN N = 6;
                                                                               00496000
                                                                               00497000
       ELSE N = TCTALSHUMSCI - POLLIMSGSCOUNT:
                                                                               00498000
```

```
MAITING1 = WAITING1 + 1;
                                                                                         00499000
     LO J = 1 TO N;
                                                                                         00500000
         POLLINSGSCOUNT = POLLSMSGSCOUNT + 1;
                                                                                         00501000
        OUTIAPLE1(TC1 +1) = O1H;
OUTIAPLE1(TC1 +2) = '0';
DC I1 = 1 TO 3;
CUITABLE1(TC1 + 2 + I1) =
FOLLTAPLE1(I1 + POLLINSG$COUNT * 4 - 4);
                                                                                         00502000
                                                                                         00503000
                                                                                         00504000
                                                                                         00505000
                                                                                         00506000
        END;
TC1 = TC1 + 6;
                                                                                         00507000
                                                                                         00508000
        OUTIABLE 1 (TC1) = 03H;
                                                                                         00508010
     ENL;
                                                                                         00509000
     MSGCCUNT1 (WAITING 1) = N + 6;
                                                                                         00510000
1 * 4/
                                                                                         00511000
144/
                                                                                         00512000
/*
            INPUT FROM HODEN
                                                                                         00513000
1001
                                                                                         00514000
                                                                                         00515000
IPOLL11: CS = CSECATA XOR OFFH;
CS3 = CS AND 04h;
IF CS3 <> C4h THEN GO TO XX;
                                                                                         00516000
                                                                                         00517000
                                                                                         00518000
CHAR = INPUTSEDATA;
                                                                                         00519000
GO TO HE;
                                                                                         00520000
XA: CS = CS AND OTH;
                                                                                         00521000
IF CS=61h THEN GO TO GC;
                                                                                         00522000
ww: IF CC > 0 THEN GO TO LCT;
                                                                                         00523000
   ELSE GO TO OPOLL1;
                                                                                         00524000
/* HOLEH INPUT */
                                                                                         00525000
GC: CHAR = INPUT3SDATA XOR OFFH;
                                                                                        00526000
CC= CC+1;
1F CC >= 2GGU THEN GO TO LCT;
                                                                                         00527000
                                                                                         00528000
CHAR: STORE (CC) = CHAR;
IF CC=1 THEN GO TC ST;
                                                                                         00529000
                                                                                        00530000
ELSE GO TO MUTT;

/* ENCRYPTION TEST */
ENCRYPTION TEST OF ENCRYP1;
                                                                                        00531000
                                                                                        00532000
                                                                                        00533000
ELSE GO TO 1CT;
                                                                                        00534000
             SCH TEST
                                                                                         00535000
ST: 11 Chak = SON THEN GO TO LCT;
1F Chak = *** THEN GO TO LCT;
                                                                                         00536000
                                                                                         00537000
IF CHAR = GAAH THEN GO TO ENCRYPT:
                                                                                         00538000
   ELSE PJI=1;
                                                                                        00539000
GO TO LC1;
                                                                                        00540000
ENCHYPT: ENC = ENC + 1;
                                                                                         00541000
GO TO LCT;
/* MPU JOB TEST */
MJI1: 1F MJ)=1 THEN GO TO LCT;
                                                                                         00542000
                                                                                        00543000
                                                                                        00544000
IF CC = 2 THEN GO TO ENCTEST;
                                                                                         00545000
IF ENC = 2 THEN OUTPUT (233) = 11h;
IF CL=3 THEN GO TO CHAR3T;
                                                                                         00546000
                                                                                        00547000
IF CC=4 THEN GO TO CHAR4T;
                                                                                         00548000
IF CC=5 THEN GO TO CHAR5T;
                                                                                        00549000
   ELSE GO TO LCT;
                                                                                         00550000
CHARST: IF CHAR . ADRI THEN GO TO LCT:
                                                                                         00551000
   ELSE MJF=1;
                                                                                        00552000
GO TO LCT;
CHAR41: IF CHAR= ADR2 THEN GO TO LCT;
                                                                                         00553000
                                                                                         00554000
   ELSE MJF=1;
                                                                                         00555000
60 16 LC1;
                                                                                        00556000
CHARST: IF CHAR= ADR3 THEN GO TO LCT;
                                                                                        00557000
ELSE MJF=1;
                                                                                        00558000
    LAST CHARACTER TEST */
                                                                                         00559000
LCT: TCLK = INPUT (236);
                                                                                         00560000
```

The second secon

```
ICLK = ICLK AND OIH;
IF ICLK = ICLKP THEN GO TO OPOLL1;
                                                                                          00561000
                                                                                          00562000
      TCLKCAT = TCLKCAT + 1;
                                                                                          00563000
      TCLKP = TCLK;
                                                                                          00564000
      IN ICLNORT < 24H THEN GO TO OFCLL1:
                                                                                          00565000
      TCLK = U;
                                                                                          00566000
      TCLKP = C;
                                                                                          00567000
      TCLFCNT = 0:
                                                                                          00566000
CCSPRESENT = CCSPHESENT + 1;
                                                                                          00569000
IF CC$FRESENT > CC THEN CC$PRESENT = 0;
ELSE GL TO OPOLL1;
                                                                                          00570000
                                                                                          00571000
    OUTFUT (.33) = 10h;
                                                                                          00572000
   ENC = 0;
                                                                                          00573000
1441
                                                                                          00574000
100/
                                                                                          00575000
/**/
/*
                                                                                          00576000
/* EDAC ALGORITHM - NOW TOM POLITING MSG'S */
EDACSECUTINE: IF EDACSFLAG <> OTH THEN GO TO HJT2;
                                                                                          00578000
      IF MJF <> 1 THEN GO TO MJT2;
                                                                                          00579000
      IF FOILSPLAG <> OTH THEN GO TO HJT2:
                                                                                          00580000
      IF TORSFLAG <> 'O' THEN GO TO MJT2;
                                                                                          00581000
      IF CC <> 7 THEN GC TO HJT2;
                                                                                          00582000
      DC J = 1 TO 7;
                                                                                          00583000
        IF CHARSSTORE (J) <> NTDRSEDACSTABLE (J)
                                                                                          00564000
            THEN GO TO RETRANSHITS ! LAGS SET:
                                                                                          00585000
                                                                                          00586000
      RETHANSMITSFLAG = OH;
REPOLLSFLAGSCOUNT = O;
                                                                                          00567000
                                                                                          00588000
      GO TO MUTZ;
                                                                                          00589000
RETRANSHITIFLAGISET: RETRANSHITEFLAG = 01H;
                                                                                          00590000
      REPCLLIFIAGSCOUNT - REPOLLEFLAGSCOUNT + 1;
                                                                                          00591000
      IF REFULLSFLAGSCOUNT > 3 THEN REPOLLSFLAGSCOUNT = 0;
                                                                                          00592000
      IF RESCLISFLAGSCOUNT = 0 THEN RETRANSMITSFLAG = OH;
                                                                                          00593000
/**/
                                                                                          00594000
  /* MODE 1EST & FORMATTING */
                                                                                          00595000
HJT2: IF HJF=1 THEN GO TO MPUSTABLESBUILD :
                                                                                          00596000
MODE = CHARISTORE (2);
                                                                                          00597000
IF MOLE = 'C' THEN GO TO CS$TEST;
                                                                                          00598000
IF MOLE = 'L' THEN GC TO CS$TEST;
IF MODE = 'E' THEN GC TO CS$TEST;
                                                                                          00599000
                                                                                          00600000
IF MODE # 'F' THEN GO TO PRESLOAD;
                                                                                          00601000
IF BCDL = 'h' THEN GU TO NRB$LOAD;
                                                                                          00602000
IF MODE = 'A' THEN GC TO HRBSLOAD;
                                                                                          00603000
IF HOLE = 'X'THEN GO TO PTT$LOAD;
                                                                                          00604000
IF HODE = 'Y' THEN GO TO RTT$LOAD;
IF HODE = 'Z' THEN GO TO NTT$LOAD;
                                                                                          00605000
                                                                                          00606000
GO TO MPUSTABLESBUILL;
                                                                                          00607000
/**/
                                                                                          00608000
144/
                                                                                          00609000
/* CHECKIN STATUS TEST */
                                                                                          00610000
CSSTEST: IF CSMODE= "1" THEN GO TO CCSTEST:
                                                                                          00611000
STARSINSERT: CHARSSTORE (1) = 121;
                                                                                          00612000
GO TO GUISTABLESBUILD;
                                                                                          00613000
   CHARACTER COUNT TEST +/
                                                                                          00614000
/* THE CC TEST REGULARS PCI/RCI/NCI CHECKINS TO HAVE */

/* A MINIMUM OF 5 CHARACTERS. MESSAGES, HONEVER, MAY */

/* BE AFPENDED AFTER THE 9TH CHARACTER. */

CCSTEST: IF CC >= 9 THEN GO TO PCISTEST;

ELSE CHARASTORE(1) = "?";
                                                                                          00615000
                                                                                          006:6000
                                                                                          00617000
                                                                                          00618000
                                                                                          00619000
GU TU OUTSTABLESBUILT;
FCISTEST: IF HODE <> *C* THEN GO TO RCISTEST;
                                                                                          30620000
                                                                                          00621000
ELSE NUMIPCI = NUMIPCI + 1;
PCISPOINTER = 4 + ( NUMIPCI - 1 ) + 1;
                                                                                          00622000
                                                                                          00623000
```

STATE OF THE PARTY OF THE PARTY

```
DO 1 = 6 TC 9;
                                                                                        00624000
   FC1$STORE (FC1$POINTER) = CHARTSTORE (1) ;
                                                                                        00625000
   PCISPGINIER = PCISPOINTER + 1;
                                                                                        00626000
END :
                                                                                        00627000
DU 1 = 0 TL 12;
OUTSTABLE (TC + 1 + 1) = PCISHEADER (I);
                                                                                        00628000
                                                                                        00629000
END:
                                                                                        00630000
NC = CC + 13;
TC = TC + 13;
                                                                                        00631000
                                                                                        00632000
GO TO CIENUMETEST;
                                                                                        00633000
ACISTEST: IF MODE <> 'D' THEN GO TO NCISTEST;
                                                                                        00634000
ELSE NUMBECT = NUMBECT + 1;
HCISPOINTER = 4 + ( NUMBECT - 1 ) + 1;
                                                                                        00635000
                                                                                        00636000
DO 1 = 6 TC 9;
                                                                                        00637000
   ACISSIONE (RCISFOINTER) = CHARSSTORE (1);
                                                                                        00636000
    RCISPCINTER = RCISPOINTER + 1:
                                                                                        00639000
END;
                                                                                        00640000
DO I = 0 TC 3;
OUTSTABLE(TC + I + 1) = RCISHEALER(I);
                                                                                        00641000
                                                                                        00642000
END;
AC = CC + 4;
TC = TC + 4;
                                                                                        00643000
                                                                                         30644000
                                                                                        00645000
GO TO CLINUESTEST:
                                                                                        00646000
ACISTEST: NUMBECI = AUMENCI + 1;
ACISECINTEN = 4 + ( NUMBECI - 1 ) + 1;
                                                                                        00647000
                                                                                        00646000
DO 1 = 6 TL 9 ;
                                                                                        00649000
    ACISTORE (NCISPOINTER) = CHARISTORE (I);
                                                                                        00650000
    NCISECINTER = NCISEOINTER + 1;
                                                                                        00651000
                                                                                         00652000
OUTSTABLE (IC + I +1) = hCISHEALER (I);
                                                                                        00653000
                                                                                        00654000
END;
AC = CC + 4;
                                                                                         00655000
                                                                                        20656000
TC = TC + 4;
                                                                                        00657000
/##/
                                                                                        00658000
144/
                                                                                        00659000
Clanumates1: IF humspc1 + numskc1 + numskc1 >= 60
                                                                                        00660000
                        THEN GO TO CINCLESCANCEL;
                                                                                        00661000
          LLSE GO TO OUTSTABLESBUILD;
                                                                                        00662000
/* All-Call CHECKIN MODE CANCEL
                                                                                        00663000
C$MODESCANCEL: IF FOLESFLAG = G1), THEN GO TO LERODESCANCEL1;
                                                                                        00664000
                  LO I1 = 1 TO 27;
                                                                                         00665000
             CUTTABLE 1 (1C1+11) = OVER 1 CONHOR (11-1);
                                                                                         00666000
                 END;
                                                                                         30667000
      TC1 = TC1 + 27;
                                                                                         00668000
TC1 = 1C1 + 27;

WAITING1 = WAITING1 + 1;

MSGCOUNTI(WAITING1) = 27,

/* CHECKIN MODE CANCEL MSG TO TERMINAL OPERATOR
C$MOLESCANCEL1: DC 1 = 1 TO 27;

OUTSTABLE (TC+1) = OVERFLCWHDR (I-1);
                                                                                         00669000
                                                                                        00670000
                                                                                        00671000
                                                                                        00672000
                                                                                        00673000
        END;
TC = TC + 27;
                                                                                        00674000
                                                                                        00675000
        MC * NC + 27;
CINCLE = 101;
                                                                                        00676000
                                                                                        00677000
        GC TC GUTSTABLES BUILD ;
                                                                                        00678000
100/
                                                                                        00679000
100/
                                                                                        00660000
HPU$1Able$ EUILD: NC = CC;
GU1$7Able$ bUILD: DO 1 = 1 TO CC;
                                                                                         00681000
                                                                                         00682000
   OUTSTABLE (TC + I) = CHARSSTORE (I);
                                                                                         00683000
10 - 10 + LC ;
                                                                                        30684000
                                                                                         00685000
halfing - halfing + 1;
                                                                                         00686000
```

The state of

```
MSGSCOUNT(WAITING) = MC;
                                                                                        00687000
CC = 0;
                                                                                        00688000
                                                                                        00689000
BJE = U:
                                                                                        00690000
OUTPUT (233) = 16H;
                                                                                        00691000
                                                                                        00692000
GO 10 OPCL11;
                                                                                        00693000
1 * */
                                                                                        00694000
100/
                                                                                        00695000
PRESIDAD: DO I = 0 TO 11;
GUTATABLE (TC + I + 1) = PRESHEADER(I);
                                                                                        00696000
                                                                                        00697000
END:
                                                                                        00698000
AC = CC + 12;
TC = TC + 12;
GO TO OUT$TABLE$BUILD;
                                                                                        00699000
                                                                                        00700000
                                                                                        00701000
00702000
REBILOAD: DO I = 0 TO 3;
OUTSTABLE(TC + I + 1) = ERBSHEALER(I);
                                                                                        00703000
END;
                                                                                        00704000
NC = CC + 4;
TC = TC + 4;
                                                                                        00705000
                                                                                        00706000
GO TO GUTSTABLESPUILL:
                                                                                        00707000
hrbsload: Lo I = 0 TO 3;
GUTSTABLE(TC + I + 1) = hrbsheauer(I);
                                                                                        00708000
                                                                                        00709000
END;
                                                                                        00710000
NC = CC + 4;
TC = TC + 4;
                                                                                        00711000
                                                                                        00712000
GO TO OUTSTABLESBUILL;
                                                                                        00713000
/**/
                                                                                        00714000
1441
                                                                                        00715000
PTT$LOAD: IF C$MODE = 'O' THEN GC IC STAHSINSERT;
                                                                                        00716000
   IF CC < 5 THEN GO TO STARSINSERT;
LO 1 = 6 TO CC EY 4;
                                                                                        00717000
                                                                                        00718000
       NUMSECI = NUMSECI + 1;
                                                                                        00719000
       FC1SFGINTER = 4 * (NUMSPCI - 1) + 1;
                                                                                        00720000
       LO J = 6 TO 3;
         FC14STORE ( PC1ffOINTER ) = CHARASTORE (1+J);
                                                                                        00722000
                                                                                        00723000
          FC18POINTER = PC18POINTER + 1;
       INL:
                                                                                        00724000
   IF NUMSEC + NUMSEC + NUMSEC >= 80 THEN GO TO PITSLOADSEND; I; (CC - I) < 4 THEN GO TO PITSLOADSEND;
                                                                                        00725000
                                                                                        00726000
  ENL:
                                                                                        00727000
PTT$LOADSEND: DO I = 0 TO 11;
                                                                                        00728000
   OUTSTABLE (TC + 1 + 1) = PTTSHEACER (I);
                                                                                        00729000
                                                                                        00730000
NC = CC + 14 ;
                                                                                        00731000
TC = TC + 14 ;
IF NUMSECT + NUMSECT + NUMSECT >= 6C THEN GO TO CSHODESCANCEL;
                                                                                        00733000
IF RUTSPEL Y HOTERES HOLLE;

GO TG GUTSTAPLESBUILE;

FITSLOAD: IF C$HODE = "0" THEN GG TC STARSINSERT;

IF CC < 9 THEN GG TO STARSINSERT;

DO 1 = 6 TO CC BY 4;
                                                                                        00734000
                                                                                        00735000
                                                                                        00736000
                                                                                        00737000
       NUMBROI = NUMBROI + 1;
                                                                                        00738000
       RCI + FOIRTER = 4 * (NUMSRCI - 1) * 1;
                                                                                        00739000
       LO J = C TO 3;
                                                                                        00740000
         hclistore( rcisfointer ) = charistore(I + J);
                                                                                        00741000
          ACISPGINTER = RCISPOINTER + 1;
                                                                                        00742000
   IF NUMSECT + NUMSECT + NUMSECT >= 80 THEN GO TO RITSLOADSENC;
                                                                                        00744000
   IF (CC - 1) < 4 THEN GO TO RTTSLCADSEND;
                                                                                        00745000
  END:
                                                                                        00746000
httsloadsent: DO I = 0 to 3;
OUTSTABLE(TC + I + 1) = httsheader(I);
                                                                                        00747000
                                                                                        00748000
                                                                                        00749000
```

Section 1

A STATE OF THE PARTY OF THE PAR

```
hC = CC + 4 ;
TC = TC + 4 ;
                                                                                00750000
                                                                                00751000
IF BURSPCI + BURSPCI + NURSECI >= 60 THEN GO TO CSHODESCANCEL;
                                                                                00752000
GO TO CUTSTABLESBUILD;
httsload: 15 cshode = 'O' Then GO TO STARSINSERT;
                                                                                00753000
                                                                                00754000
   IF CC < 5 THEM GO TO STARSINSERT;

LO I = 6 TO CC BY 4;

NUMSECI = NUMSECI + 1;
                                                                                00755000
                                                                                00756000
                                                                                00757000
        NCISFOINTER = 4 * (NUMSNCI - 1) + 1;
                                                                                00758000
      DO J = 0 TO 3;
    NCISSTORE( NCISPOINTER ) = CHARSSTORE(I+J);
    NCISPOINTER = NCISPOINTER + 1;
                                                                                00759000
                                                                                00760000
                                                                                00761000
      ENL:
                                                                                00762000
   IF NUMSECT + NUMSECT + NUMSECT >= 80 THEN GO TO NTT$LOADSEND;
                                                                                00763000
   IF (CC -1) < 4 THEN GO TO NTTSLCADSEND;
                                                                                00764000
  END:
                                                                                00765000
ATTSLOADJEND: DO I = 0 TO 3;
                                                                                00766000
   OUTSTABLE (TC + I + 1) = NTTSHEADER (I) ;
END:
                                                                                00768000
NC = CC + 4;
                                                                                00769000
TC = TC + 4;

1F NUMSPCI + NUMSRCI + NUMSNCI >= 86 THEN GO TO CSMODESCANCEL;
                                                                                00770000
                                                                                00771000
GO TO CUTSTABLESBUILD;
                                                                                00772000
00773000
                                                                                00774000
                                                                                00775000
         IF IC > 0 THEN GO TO ECHSTEST;
                                                                                00776000
            ELSE GO TO IPOLL161;
                                                                                00777000
EONSTEST: 11 OUTSCOUNT < HSGSCOUNT(1) THEN GO TO SPACESTEST;
    ELSE DELAY = 1;
SPACESTEST: IF RS2328FLAG = 01H THEN GO TO RS232$SPACESTEST;
                                                                                00780000
SPACE = INFLISSDATA XOR OFFH;
                                                                                00781000
SPACE = SPACE AND C2H;
IF SPACE = C2H THEN GO TO OUT$CHAK;
                                                                                00762000
                                                                                00783000
   ELSE DELAY = 0;
                                                                                00764000
60 TO 11011101;
                                                                                00785000
                                                                                00786000
hs.32ISPACERTEST: SPACE = IMPUT(237);
                                                                                00767000
                    SFACE = SPACE AND OTH;
                                                                                00768000
                    IF SPACE = 01h THEN GO TO OUTSCHAR;
                                                                                00769000
                       ELSE DELAY = C;
                                                                                00790000
                    GO TO IPOLL101;
                                                                                00791000
                                                                                00792000
LELAYTHOUTINE: CLK = INPUT (230);
                                                                                00793000
     CLK = CLK AND OTH;
IF CLK = C1KF THEN GO TO IPOL1101;
                                                                                00794000
                                                                                00795000
     CLKCNT = CLKCNT + 1;
                                                                                20796000
     CIKE = CLK;
                                                                                00797000
     IF CLECKT < BUFFERSDELAY THEN GO TO IPOLLIGI;
                                                                                00798000
     CUTFUT(235) = 6h;
                                                                                00799000
     IF CLACAT < HSGSDELAY THEN GO TO IPOLLIU1;
                                                                                00800000
     CELAY = C;
                                                                                00661000
     CLKCNT = G;
                                                                                00802000
     GO TO 1POLL 101;
                                                                                00803000
100/
                                                                                00604000
/* OUTPUT TO MPU */
                                                                                00805000
                                                                                00606000
GUTCHAR: IF HS232$FLAG + OTH THEN GC TO RS232$OUTCHAR;
                                                                                00807000
OUTFUT(235) = GIN;
OUTFUT1sDATA = OUTSTABLE(OUTSCOUNT) NOW OFFH;
                                                                                00808000
                                                                                00809000
GUISCOUNT = OUTICOUNT + 1;
                                                                                00610000
GO TO SHIFTLUBN;
                                                                                00611000
100/
                                                                                00812000
```

*

A selected to the selection of

```
RS232SOUTCHAR: OUTPUT (237) = 25H;
OUTPUT (236) = OUTSTAFLE (OUTSCOUNT);
OUTSCOUNT = OUTSCOUNT + 1;
                                                                                        00813000
                                                                                         00814000
                                                                                         00815000
SHIFTDC=N: IF OUTSCOUNT <= MSG$COUNT(1) THEN GO TO IPOLL 101;
                                                                                         00816000
          CUIFUT (237) = 15H;
                                                                                         00817000
          DO 1 = OUTSCOUNT TO TC ;
                                                                                         00818000
           OUTSTABLE ( 1 + 1 - OUTSCOUNT ) = OUTSTABLE ( I ) ;
                                                                                         00819000
          ENL;
                                                                                         00820000
LO 1 = 1 TC MAITING ;
                                                                                         00821000
   MSG$COUNT(I) * MSG$COUNT(I+1) ;
                                                                                         00822000
ENC;
TC = TC - ( CUTSCOUNT -1 ) ;
                                                                                         00823000
                                                                                         00824000
HSGCOURT (WAITING) = C;
WAITING = WAITING - 1;
                                                                                         00825000
                                                                                         00826000
OUTSCOUNT = 1;

IF NAITING = C THEN TC = 0;

/* TX & NX HAJON LOOP - MPU */
                                                                                        00827000
                                                                                        00628000
                                                                                        00829000
1 * */
                                                                                        00830000
1441
                                                                                        00631000
                                                                                         00832000
IPULL 101: IF RS232#FLAG = OTH THEN GO TO RS232#INPUTT;
                                                                                        00633000
                ELSE GO TO IPOLL 1011:
                                                                                        00834000
                                                                                         00835000
h5232$1NFU11: CS1 = 1NPUT(237);
CS1 = CS1 AND 02H;
                                                                                         00836000
                                                                                         00£37000
                 IF CS1 = 02H THEN CHALL = INPUT (236);
                                                                                        00086900
                     ELSE GO TO IPOLL1611;
                                                                                         00639000
                 60 TO h52328GC1;
                                                                                         00640000
1 * * /
                                                                                         00641000
IPOLLIGIT: CS1 = CS1EDATA XOR OFFH;
                                                                                        00642000
CS4 = CS1 AND G4H;
                                                                                        0084-3000
II CS4 <> GAH THEN GO TO YY;
                                                                                         00844000
CHART = INFUTIOUSDATA;
                                                                                         00845000
GO 10 22;
YY: CS1 = CS1 AND OTH;
                                                                                        00846000
                                                                                         00847000
IF CS1 = 01h THEN GG TO GC1;
                                                                                         00848000
22: IF CC1 > 6 THEN GO TO LCT1;
                                                                                         00849000
  ELSE GO 10 OPOLL101:
                                                                                         00850000
100/
                                                                                         00851000
/*
         INFUT FROM MPU
                                                                                         00052000
/**/
                                                                                         00853000
GC1: CHAR1 = INFUT103$DATA XOR OFFH;
                                                                                        00854000
h5232$GC1: CC1 = CC1 + 1;

IF CC1 >= 2000 THEN GO TO LCT1;
                                                                                         00855000
                                                                                         00856000
CHARSTORET (CC1) = CHAR1;
                                                                                         00857000
/* */
/* header test */
                                                                                         00858000
                                                                                         00659000
/* */
IF CC1 <= 3 THEN GO TO LCT1;
                                                                                         00860000
                                                                                         00861000
IF CC1 = 4 THEN GO TO CHAR4T1;
IF CC1 >= 5 THEN GO TO LCT1;
CHAR4T1: IF CHAR1 = 21H THEN OUT1 = 0;
                                                                                         00662000
                                                                                         00863000
                                                                                         00864000
             ELSE GO TO OUTFLAGT;
                                                                                         00865000
          GC TC LCT1;
                                                                                         00866000
CUTILAGI: OUT1 = 1:
                                                                                         00667000
100/
                                                                                         00868000
/*
        LAST CHARACTER TEST
                                                                                         00869000
/**/
                                                                                         00870000
LCT1: TCLK1 = INPUT (230);
                                                                                         00871000
       TCLK1 = TCLK1 AND OTH;

IF TCLK1 = TCLK1 THEN GO TO CPOLLIO1;

TCLKCNT1 = TCLNCNT1 + 1;
                                                                                         00672000
                                                                                         00673000
                                                                                        00674000
       TCLKF1 = TCLK1;
                                                                                        00875000
```

A STATE OF THE STA

```
00876000
       IF TCLKCNT1 < MSGSOVERSDELAY THEN GO TO OPOLL101;
                                                                                          00877000
       1C1K1 = 0:
       1CLK # 1 = 0;
                                                                                          00878000
                                                                                          00679000
       ICLKCKT1 = 0:
       CC 15 FRESENT = CC 15 PRESENT + 1;
                                                                                          00000000
       IF CC18FRESENT > CC1 THEN CC18PRESENT = 0;
                                                                                          00881000
       ELSE GO TO OPOLL101:
                                                                                          00882000
/**/
                                                                                          00883000
144/
                                                                                          00884000
/* */
/* */
                                                                                          00885000
                                                                                          00886000
/* CUTPUT COMPOSITION */
                                                                                          00887000
                                                                                          00889000
OUTPUTSCONFOSE: IF OUT! <> ! THEN GC TO COMPOSE!;
                                                                                          00690000
  DO CASE EATSMSG$FLAG1;
                                                                                          00891000
      GC TO MUDERSCUTPUTSCOMPOSE1;
                                                                                          00892000
      IF FOILFFLAG = OTH THEN GO TO EATSMSG1;
ELSE GO TO MODEMSOUTFUTSCOMPOSE1;
                                                                                          00893000
                                                                                          00894000
      GC TO LATSMSG1;
                                                                                          00895000
                                                                                          00896000
MOLENICUTPULICONPOSE1: IF CHARSTORE 1(4) = *** THEN
                                                                                          00897000
                                GO TO MODEMSOUTPUTSCOMPOSE2;
                                                                                          00898000
  50 I1 = 1 TO CC1;
                                                                                          00899000
OUTTAble 1(TC1 + I1) = CHARSTORE 1(I1);
                                                                                          00900000
                                                                                          00901000
TC1 = TC1 + CC1;
WAITING1 = WAITING1 +1;
                                                                                          00903000
MSGCGUNT1(WAITING1) = CC1;
                                                                                          00904000
EATSMSG1: CC1 = 0;
                                                                                          00905000
GUT1 - 0:
                                                                                          00906000
GO TO OPOLITOT:
                                                                                          00907000
/**/
/* LOOP-AROUND CUTFUT MESSAGE COMPOSITION */
                                                                                          00908000
                                                                                          20909000
144/
                                                                                          00910000
RODER$GU1PU1$COMPCSE2: DO 11 = 4 TO CC1;
CUTTABLE1(TC1 + 11 - 3) = CHARSTORE1(11);
                                                                                          00911000
                                                                                          00912000
                            END;
                                                                                          00913000
              TC1 = TC1 + CC1 ;
                                                                                          00914000
              WAITING1 = WAITING1 + 1;
                                                                                          00915000
              MSGCOUNT1(WAITING1) = CC1;
                                                                                          00916000
              cc1 . 0:
                                                                                          00917000
              GUT1 = C:
                                                                                          00918000
              GO TO OPOLLIO1:
                                                                                          00919000
                                                                                          00920000
/* CUMPOSITION PROGRAM */
                                                                                          00921000
                                                                                          00922000
/* TEST HOLE CHARACTER */
                                                                                          00923000
                                                                                          00924000
COMPOSET: IF CCT < & THEN GO TO MOLLERRORT;
                                                                                          00925000
IF CHARSTORE 1(L) <> ADRI THEN GO TO ADRERROR1; IF CHARSTORE1(7) <> ADR2 THEN GO TO ADRERROR1;
                                                                                          00926000
                                                                                          00927000
IF CHARSTORET(b) <> ADR3 THEN GO TO ADRERRORT;
                                                                                          30928000
                                                                                          00929000
/* COMSUP HOLE RECOGNITION CHARACTERS
                                                                                          00930600
IF CHARSTORE 1(5) = "A" THEN GO TO GCIRESET;
IF CHARSTORE 1(5) = "E" THEN GO TO GT1;
IF CHARSTORE 1(5) = "C" THEN GO TO FT1;
                                                                                          00932000
                                                                                          00933000
                                                                                          00934000
  IF CHARSTORE 1(5) = 'D' THEN GC TC RT1;
IF CHARSTORE 1(5) = 'E' THEN GC TC NT1;
IF CHARSTORE 1(5) = 'F' THEN GC TC PCIRESET;
IF CHARSTORE 1(5) = 'G' THEN GC TC RCIRESET;
                                                                                          00935000
                                                                                          00936000
                                                                                          00937000
                                                                                          00938000
```

Sec. 3. 1.

A Section ! Patient

```
IF CHARSTORE1(5) = "h" THEN GO TO NCIRESET;
IF CHARSTORE1(5) = "I" THEN GO TO ADRCHANGE;
IF CHARSTORE1(5) = "J" THEN GO TO POLLSHODE;
                                                                                                                 00939000
                                                                                                                 00940000
                                                                                                                 00941000
  IF CHARSTORE! (5) = 'N' THEN GO TO POLLSHODE;

IF CHARSTORE! (5) = 'N' THEN GO TO NONSTDHSDELAYSSET;

IF CHARSTORE! (5) = 'L' THEN GO TO POLLSHITERUPT;

IF CHARSTORE! (5) = 'N' THEN GO TO POLLTABLEBUILD!;

IF CHARSTORE! (5) = 'N' THEN GO TO CHECKINSDELETE;

IF CHARSTORE! (5) = 'O' THEN GO TO SLOTSREASSIGN;

IF CHARSTORE! (5) = 'P' THEN GO TO EDACSHODESSET;
                                                                                                                 00942000
                                                                                                                 00943000
                                                                                                                 00944000
                                                                                                                 00945000
                                                                                                                 00946000
                                                                                                                 00947000
    IF CHARSTORE 1(5) = 'Q' THEN GO TO RS232$ NODE$SET;
                                                                                                                 00948000
IPLATEST: IF CHARSTONE (5) = "R" THEN GO TO PROGRAMSSTART;
                                                                                                                 00949000
   IF CHARSICREI(5) = 'S' THEN GO TO CHECKINHODESET1;
IF CHARSICREI(5) = 'T' THEN GO TO TTR1;
IF CHARSICREI(5) = 'U' THEN GO TO TBLDUMP1;
                                                                                                                 00950000
                                                                                                                 00951000
                                                                                                                 00952000
IF CHARSTONE1(5) = 'V' THEN GO TO EATSASGSSET;
IF CHARSTONE1(5) = 'X' THEN GO TO PITR1;
                                                                                                                 00953000
IF CHARSTORE 1(5) = 'Y' THEN GO TO HITRI;
IF CHARSTORE 1(5) = 'Z' THEN GO TO HITRI;
IF CHARSTORE 1(5) = '1' THEN GO TO PITSLOAD 1;
IF CHARSTORE 1(5) = '2' THEN GO TO HITSLOAD 1;
IF CHARSTORE 1(5) = '3' THEN GO TO HITSLOAD 1;
                                                                                                                 00935000
                                                                                                                 00956000
                                                                                                                 00957000
                                                                                                                 00958000
                                                                                                                 00959000
                                                                                                                 00960000
   LISE GC TC MODEERROR1;
/**/
                                                                                                                 00961000
 /++/
                                                                                                                 00962000
SLOTSREASSIGN: IF CC1 < 22 THEN GO TO MODEERROR1;
                                                                                                                 00963000
    IF CHARSTORET (21) > 35H THEN GO TO MODEERROR1;
                                                                                                                 00964000
     IF CHARSTORE 1 (21) < 30H THEN GO TO MODEERROR 1;
                                                                                                                 00965000
     IF CHARSTORE 1 (22) > 39H THEN GO TO MODERROR 1;
                                                                                                                 00966000
     IF CHARSTONE 1 (22) < 30h THEN GO TO MODERROR 1;
    TX$SLOT$INITIALH = (CHARSTORE 1(21) - 30H) + 10;
TX$SLOT$INITIALL = CHARSTORE 1(22) - 30H;
IF TX$SLOT$INITIALH + TX$SLOT$INITIALL > S4 THEN GO TO HODEERROR1;
IF TX$SLOT$INITIALH + TX$SLOT$INITIALL = 0 THEN GO TO HODEERROR1;
                                                                                                                 00968000
                                                                                                                 00969000
                                                                                                                 00970000
                                                                                                                 00971000
     TDMSPOLLSTXSSLOTSINITIAL = TXSSLCTSINITIALH + TXSSLOTSINITIALL;
                                                                                                                 00972000
    TEMSPOLISTASSLOT = TOMSPOLISTASSLOTSINITIAL;
                                                                                                                 00973000
      TDMSFOLLSTXSSIOTSINITIAL IS LIMITED TO A KANGE OF 1 TO 54 +/
                                                                                                                 00974000
         POLL>SLOTIH = CHARSTORE1(9);
                                                                                                                 00975000
          FOLLESLOTIL - CHARSTORE (16);
                                                                                                                 00976000
          FOLLISLOT2H = CHARSTORE1(11);
                                                                                                                 00977000
          POILISICT2L = CHARSTORE1(12);
                                                                                                                 00978000
          FOLLESLOTSH = CHARSTORE1(13);
          POLLISLOT31 = CHARSTORE1(14);
          POLLESLOTAN = CHARSTORE1(15);
                                                                                                                 00961000
          POILSSLOT41 = CHARSTORE1(16);
                                                                                                                 00982000
          POLLISIOTSH = CHARSTORE (17)
                                                                                                                 00983000
          FOLLISLOTS1 = CHARSTORE 1 (18);
                                                                                                                 00048600
          FOLLESLOTEH = CHARSTORE1(19);
                                                                                                                 00985000
         FOILESLOTEL = CHARSTORE 1(20);
                                                                                                                 00986000
          GO TC COMSUPLOAD1;
                                                                                                                 00987000
/**/
                                                                                                                 00968000
AS232$MODE$SET: IF CC1 < 9 THEN GC TO MODEERGR1;
IF CHARSTORE1(9) = '0' THEN RS232$FLAG = OH;
IF CHARSTORE1(9) = '1' THEN RS232$FLAG = O1H;
                                                                                                                 00989000
                                                                                                                 00991000
                                                                                                                 00992000
              GC TO COMSUFLOAD1:
/**/
                                                                                                                 00993000
                                                                                                                 00994000
/**/
LATSHSGSSET: IF CC1 < 9 THEN GO TO PODEERROR;
IF CHARSTORE1(9) = '0' THEN EATSHSGSSLAG1 = OH;
IF CHARSTORE1(9) = '1' THEN EATSHSGSFLAG1 = OH;</pre>
                                                                                                                 00995000
                                                                                                                 00996000
                                                                                                                 00997000
              IF CHARSTORE 1(9) = '2' THEN EATSHSG$FLAG1 = 02H;
                                                                                                                 00998000
              CHARSTORE 1(9) = EATSMSGSFLAG1 + 30H;
                                                                                                                 00999000
              GO TO COMSUPLOAD1;
                                                                                                                 01000000
                                                                                                                 01001000
```

A Transfer of the Park of the

```
01002000
EDACSFICELSSET: IF CC1 < 9 THEN GO TC ADRERROR1;
IF CHARSTORE1(9) = '1' THEN EDACSFLAG = 01H;
IF CHARSTORE1(9) = '0' THEN EDACSFLAG = OH;
                                                                                                                                                                                     01003000
                                                                                                                                                                                     01004000
                                                                                                                                                                                     01005000
                  GC TC COMSUFICADI:
                                                                                                                                                                                     01006000
/**/
                                                                                                                                                                                      01007000
                                                                                                                                                                                     01008000
 ADRCHANGE: IF CC1 < 11 THEN GO TO ADRERROR1:
                                                                                                                                                                                     01009000
                            ADE1 = CHARSTORE1(9):
                                                                                                                                                                                      01010000
                            ADR2 = CHARSTORE 1 (10);
                                                                                                                                                                                      01011000
                            ADR3 = CHARSTORE1(11);
                                                                                                                                                                                      01012000
                            GO TO CONSUPLOAD1:
                                                                                                                                                                                      01013000
144/
                                                                                                                                                                                      01014000
 /44/
                                                                                                                                                                                     01015000
POLLSHODE: IF CC1 < 10 THEN GO TO ADRERROR1;

IF CHARSTORE1(9) = '0' THEN POLLSFLAG = OH;

IF CHARSTORE1(9) = '1' THEN POLLSFLAG = O1H;

* INITIALIZATION OF POLLING MODE */
                                                                                                                                                                                      01016000
                                                                                                                                                                                      01017000
                                                                                                                                                                                      01018000
                                                                                                                                                                                      01019000
             IF CHARSTORE (9) = '1' THEN POLLS HSG SCOUNT = 0;
IF CHARSTORE (9) = '1' THEN
                                                                                                                                                                                      01020000
                                                                                                                                                                                      01021000
                                IDMSPOLISTXSSIOT = TDMSPOLLSTXSSLOTSINITIAL;
                                                                                                                                                                                      01022000
            IF CHARSTORE 1 (9) = '1' THEN REPOLLSTAGS COUNT = 0;
IF CHARSTORE 1 (9) = '1' THEN RETRANSMITS FLAG = OH;
IF CHARSTORE 1 (10) = 'S' THEN AUTOSREPOLLS FLAG = OH;
IF CHARSTORE 1 (10) = 'N' THEN AUTOSREPOLLS FLAG = OH;
                                                                                                                                                                                      01023000
                                                                                                                                                                                      01024000
                                                                                                                                                                                      01025000
                                                                                                                                                                                      01026000
                            GO TO CONSUPLOADI;
                                                                                                                                                                                      01027000
1 * * /
                                                                                                                                                                                      01028000
 /++/
                                                                                                                                                                                      01029000
FOLLSINTERRUFT: IF CC1 < 10 THEN GO TO ADRERRURT;
                                                                                                                                                                                      01030000
               IF CHARSTORE 1(9) = '1' THEN ATLM $1011$ INTERRUPT$ FLAG = 01H;
                                                                                                                                                                                      01031000
                                                                                                                                                                                      01032000
                                        IF CHARSTORE 1 (9) = '0' THEN
                                                                                                                                                                                      01033000
               ATCHIPOLISINTERRUPTSFLAG = OH;
                                        IF CHARSTORE 1 (10) = '1' THEN
               IDMS FOLLS INTEREUPTS FLAG = 01H;
IF CHARSTORE 1(10) = '0' THEN
                                                                                                                                                                                      01036000
                                                                                                                                                                                     01037000
               TEMSFOLLSINTERRUPTSFLAG = CH;
                                                                                                                                                                                     01038000
               CHARSTORE 1 (9) = NTDM $ POLLS INTERRUPT $ FLAG + 30H;
CHARSTORE 1 (10) = TDM $ POLLS INTERRUPT $ FLAG + 30H;
                                                                                                                                                                                      01039000
                                                                                                                                                                                      01040000
               GO TC COMSUPLOAD1:
                                                                                                                                                                                      01041000
/**/
                                                                                                                                                                                      01042000
                                                                                                                                                                                      01043000
NONSTONSCELAYSSET: IF CC1 < 10 THEN GO TO ADRERROR1;
                                                                                                                                                                                      01044000
               IF CHARSTORE 1(9) < 30H THEN GC TO MODEERROR1; IF CHARSTORE 1(9) > 39H THEN GC TO MODEERROR1;
                                                                                                                                                                                      01045000
                                                                                                                                                                                      01046000
               IF CHARSTORE 1 (10) < 30H THEN GO TO MODEERROR 1;
                                                                                                                                                                                      01047000
              IF CHARSTORE 1(10) > 39H THEN GO TO HODEERROR 1;
                                                                                                                                                                                      01048000
     /* CHARACTERS 9 & 10 ARE DELAY ENTERED IN SECONDS
                                                                                                                                                                                      01049000
                            MAXIMUM DELAY IS 99 SECONDS
                                                                                                                                                                                      01050000
               ATEMEELAYH = CHARSTORE1(9) - 30H;
ATEMEELAYL = CHARSTORE1(10) - 30H;
                                                                                                                                                                                      01051000
                                                                                                                                                                                      01052000
DELAYSCAL: NONSTDHSMSG$DELAY1 = (NTENDELAYH+3COH) + (NTDMDELAYL+3OH);
GO TC COMSUPLOAD1;
                                                                                                                                                                                      01053000
                                                                                                                                                                                      01054000
/**/
                                                                                                                                                                                      01055000
/++/
/* ACH ENROR MESSAGE
                                                                                                                                                                                      01056000
                                                                                                                                                                                      01057000
ADMERROR HESSAGE 4/
ADMERR
                                                                                                                                                                                      01058000
                                                                                                                                                                                      01059000
                            END;
                                                                                                                                                                                      01060000
                            WAITING = WAITING + 1;
                                                                                                                                                                                      01061000
                            TC = TC + 30;
                                                                                                                                                                                      01062000
                            GUITABLE (TC +1) = ODH;
CUTTABLE (TC +2) = OAH;
                                                                                                                                                                                      01063000
                                                                                                                                                                                      01064000
```

A . See Made at

```
CUTTABLE (TC+ 3) = ADR1;
                                                                                   01065000
            CUITABLE (TC +4) = ADR;
CUITABLE (TC +5) = ADR3;
CUITABLE (TC +6) = O3H;
                                                                                   01066000
                                                                                   01067000
                                                                                   01068000
             CUTTABLE (TC +7) = 03H;
                                                                                   01069000
                                                                                   01070000
             1C = TC + 7;
             ESGCOUNT (MAITING) = 37;
                                                                                   01071000
             GO TO COMSUPLOADI:
                                                                                   01072000
/**/
                                                                                   01073000
                                                                                   01074000
/* ALLITIONS TO CI TABLES */
                                                                                   01075000
100/
                                                                                   01076000
       CHECKIN TABLE TRANSFERS FROM PPU BY COMSUP COMMAND
                                                                                   01077000
                                                                                   01078000
FTT$LOAD 1: 1F CC1 < 12 THEN GO TO HODERROW 1;
                                                                                   01079000
      IF NUMSPER + NUMSREE + NUMSPER >= 80 THER GO TO OVERFLOWT;
                                                                                   01080000
      LC 1 = 9 TO CC1 BY 4;
NUMSPCI = NUMSPCI + 1;
                                                                                   01081000
                                                                                   01082000
         FC1SPOINTER = 4 * (NUMSPCI - 1) + 1;
                                                                                   01083000
         10 J = 0 TC 3;
                                                                                   01084000
             FCISSTORE (PCISPOINTER ) *
                                                                                   01085000
                 CHARSTORE 1 (1 +J);
                                                                                   01086000
            ICISPOINTER - PCISPOINTER + 1;
                                                                                   01057000
         ENL;
                                                                                   01088000
         IF humbpel + numbrel + numbrel >= 80 Then GO TO TTSLOADSENET; IF (CC1 - I) < 4 Then GO TO TTSLOADSENDT;
                                                                                   01089000
                                                                                   01090000
       END:
                                                                                   01091000
       GO TC COMSUPLOAD1;
                                                                                   01092000
ATTS LOAD 1: 1F CC1 < 12 THEN GO TO MODERROW1;
IF NUMSPC1 + NUMSRC1 + NUMSRC1 >= 80 THEN GO TO OVERFLOW1;
                                                                                   01093000
                                                                                   01094000
      DO 1 = 9 TO CC1 By 4;
                                                                                   01095000
         AUMINCI = NUMSECI + 1;
                                                                                   01096000
         HCITPOINTER = 4 * (NUMSRC1 - 1) + 1;
                                                                                   01097000
         LO J = 0 TO 3;
                                                                                   01098000
            ACI$STORE (RCI$POINTER ) =
                                                                                   01099000
                  CHARSTORET (I +J);
                                                                                   01100000
             ACISPOINTER * ACISPOINTER * 1;
                                                                                   01101000
         ENC;
                                                                                   01102000
         IF NUMSPCI + NUMSPCI + NUMSPCI >= 60 THEN GO TO TT$LOAD$END1; IF (CC1-1) < 4 THEN GO TO TT$LOAD$END1;
                                                                                   01103000
                                                                                   01104000
       END:
                                                                                   01105000
       GO TC COMSUPLOAD1:
                                                                                   01166000
ATTSLOAD 1: IF CC1 < 12 THEN GO TO HODEERROR1;
                                                                                   01107000
      IF NUMSPOX + NUMSROX + NUMSROX >= 80 THEN GO TO OVERFLOWT;
                                                                                   01108000
      DO 1 = 9 TO CC1 BY 4;
                                                                                   01109000
         AUMINCI - NUMSHCI + 1;
                                                                                   01110000
         ACISPOINTER = 4 * (NUMSHCI - 1) + 1;
                                                                                   01111000
         CO J = 0 TO 3;
                                                                                   01112000
             ACISSTORE (NCISPOINTER ) .
                                                                                   01113000
                  CHARSTORE1(I +J);
                                                                                   01114000
            ACISPOINTER = NCISPOINTER + 1;
                                                                                   01115000
         ENC;
                                                                                   01116000
         IF NUMSPEE + NUMSREE + NUMSBEE >= 80 THEN GO TO TTSLOADSENET;
                                                                                   01117000
         IF (CC1 -1) < 4 THEN GO TO TTSLOADSENU1;
                                                                                   01118000
ENT;
GO TG COMSUPLOAD1;
TTSLGADSENE1: IF NUMSPCI + NUMSRCI >= 80 THEN GO
                                                                                   01119000
                                                                                   01120000
                                                                                   01121000
                 TO OVERFLOW 1:
                                                                                   01122000
                     ELSE GO TO COMSUFICADI:
                                                                                   01123000
                                                                                   01124000
                                                                                   01125000
   FRIORITY */
                                                                                   01126000
                                                                                   01127000
```

A CONTRACTOR OF THE PARTY OF

```
01128000
/+ +/
PT1: IF NUMSPCI + NUMSRCI + NUMSNCI >= 80 THEN GO TO
                                                                                 01129000
                                                                                 01130000
      OVERFLOW1:
                                                                                 01131000
NUMSPCI = NUMSPCI + 1:
PCI$POINTER = 4 * ( NUMSPCI - 1 ) + 1;
DO I = 9 TC 12;
                                                                                 01132000
                                                                                 01133000
PCISTORE (PCISPOINTER) = CHARSTORE 1 (I);
                                                                                 01134000
                                                                                 01135000
PCISFOINTER = PCISPOINTER + 1;
ENU;
                                                                                 01136000
                                                                                 01137000
GO TO CONSUFLOAD1;
                                                                                 01138000
/* */
1 * 1
                                                                                 01139000
                                                                                 01140000
/* HOUTINE */
                                                                                 01141000
/* */
/* */
                                                                                 01142000
AT1: IF NUMSPOI + NUMSROI + NUMSACI >= 80 THER GO TO
                                                                                 01143000
       GVERFLOW 1;
                                                                                 01144000
NUMSECI = NUMSECI + 1;
                                                                                 01145000
hclsfolnteh = 4 * ( NUMSRCI - 1 ) + 1;
                                                                                 01146000
LO I = 9 TC 12;
                                                                                 01147000
  hCl3STCRE(hCl3FOINTER) = CHARSTORE1(I);
                                                                                 01148000
  HCISFOINTER = HCISFOINTER + 1;
                                                                                 01149000
                                                                                 01150000
ENL:
                                                                                 01151000
GO TO COMSUFLOADI:
1 * */
                                                                                 01152000
                                                                                 01153000
14 4/
                                                                                 01154000
/* NO TRAFFIC */
/* */
                                                                                 01155000
                                                                                 01156000
10 1/
                                                                                 01157000
NT1: IF NUMSPOI + NUMSPOI + NUMSPOI >= 80 THEN GO TO
                                                                                 01158000
       OVERFLOR1;
NUMSACI = NUMSACI + 1;
                                                                                 01159000
                                                                                 01160000
hcissointeh = 4 + (humshci - 1) + 1;
                                                                                 0 1 16 1000
LO I = 9 TC 12;
  ACISSTORE (NCISPOINTER) = CHARSTORET (I);
                                                                                 01162000
  ACISPOINTER = NCISPOINTER + 1;
                                                                                 01163000
LNL;
                                                                                 01164000
                                                                                 01165000
GO TO COMSUFLOADI;
                                                                                 01166000
/*4/
1001
                                                                                 01167000
/*
                                                                                 01166000
        GROUP CHECKIN
                                */
                                                                                 01169000
GT1: IF CC1 < 11 THEN GO TO ADRESSCHI; IF NUMBGCI >= 64 THEN GO TO OVERFLUNZ;
                                                                                 01170000
                                                                                 01171000
11 m=...
h = G;
DO I = 1 TC 16;
hUH$GCI = NUH$GCI + 1;
resoluter = 4+(NUH$G
                                                                                 01172000
                                                                                 01173000
                                                                                 01174000
                                                                                 01175000
       GCISFOINTER = 4* (NUMSGCI -1) +1;
       GCISSIORE (GCISPOINTER) = CHARSTORE 1 (9);
GCISSIORE (GCISPOINTER + 1) = CHARSTORE 1 (16);
                                                                                 01176000
                                                                                 01177000
                                                                                 01178000
       LO CASE N:
                                                                                 01179000
          GCISSTORE (GCISPOINTER + 2) = "0";
          GCISSTORE (GCISPOINTER + 2) = '1';
                                                                                 01180000
          GC13STORE (GC13POINTER + 2) = '2';
                                                                                 01181000
          GCISSTORE (GCISPOINTER + 2) = '3';
                                                                                 01182000
          GCISSTORE (GCISFOINTER + 2) = "4";
                                                                                 01183000
          GC1$STORE (GC1$POINTER + 2) = '5';
                                                                                 01184000
          GCISSTORE (GCISPOINTER + 2) = "6";
                                                                                 01185000
          GCISSTORE (GCISPOINTER + 2) = '7';
                                                                                 01186000
          GCISSTORE (GCISPOINTER + 2) = '0';
                                                                                 01187000
          GCISSTORE (GCISPOINTER + 2) = '9';
                                                                                 01188000
          GCISSTORE (GCISPOINTER + 2) = 'A';
                                                                                 01169000
          GC1$STORE (GC1$POINTER + 2) = 'B';
                                                                                 0 119 0000
```

```
GCISSTORE (GCISPOINTER + 2) = 'C';
GCISSTORE (GCISPOINTER + 2) = 'D';
GCISSTORE (GCISPOINTER + 2) = 'E';
GCISSTORE (GCISPOINTER + 2) = 'F';
                                                                                             01191000
                                                                                             01192000
                                                                                             01193000
                                                                                             01194000
       END;
                                                                                             01195000
        N = N + 1;
                                                                                             01196000
         GCIISTORE (GCISPOINTER + 3) = CHARSTORE1 (11);
                                                                                             01197000
        GCISTOINTER = GCISPOINTER + 5;
IF NUMSGCI >= 64 THEN GO TO CVERFLOW2;
                                                                                             01198000
                                                                                             01199000
END:
                                                                                             01200000
GO TO COMSUPLOAD1;
                                                                                             01261000
144/
                                                                                             01202000
100/
                                                                                             01203000
CHECKINSDELETE: IF CC1 < 11 THEN GO TO ADBERROR1:
                                                                                             01204000
                                                                                             01205000
       IF CHARSTORE 1(9) = 'P' THEN N = 0;
IF CHARSTORE 1(9) = 'R' THEN N = 1;
IF CHARSTORE 1(9) = 'N' THEN N = 2;
IF CHARSTORE 1(9) = 'G' THEN N = 3;
                                                                                             01206000
                                                                                             01207000
                                                                                             01208000
                                                                                             01209000
                                                                                             01210000
       IF N = 4 THEN GO TO MODEERACRI;
                                                                                             01211000
       IF CHARSTORE 1(10) - 30 H > 5 THEN GO TO MODEERROR1;
IF CHARSTORE 1(11) - 30 H > 9 THEN GO TO MODEERROR1;
                                                                                             01212000
                                                                                             01213000
                                                                                             01214000
                                                                                             01215006
     J = (CHARSTORE1(10) - 30H) + 10 + (CHARSTORE1(11) - 30H);
                                                                                             01216000
       DO CASE N;
                                                                                             01217000
         IF J > NUMSPOI THEN GO TO MCDEERROR1;
                                                                                             01218000
          IF J > NUMIRCI THEN GO TO MCDEERROR1;
IF J > NUMIRCI THEN GO TO MCDEERROR1;
                                                                                             01219000
                                                                                             01220000
          IF J > NUMSGCI THEN GO TO MCDEERROR1;
                                                                                             01221000
                                                                                             01222000
       END:
                                                                                             01223000
       DO CASE N;
                                                                                             01224000
           IF NUMSPCI = 0 THEN GO TO MODEERROR 1;
                                                                                             01225000
           IF NUMSRCI = 0 THEN GO TO MODEENRON!;
IF NUMSRCI = 0 THEN GO TO MODEERROR!;
                                                                                             01226000
                                                                                             01227000
           IF NUMSCCI = 0 THEN GO TO MODEENROK1;
                                                                                             01228000
                                                                                             01229000
                                                                                             01230000
       LO CASE N;
DG I = 4*J+1 TO 4*NUM$PCI;
                                                                                             01231000
                                                                                             01232000
                                                                                             01233000
               PCI$STORE(I-4) = PCI$STCRE(I);
           EAD;
                                                                                             01234000
           DO I = 4*J+1 TO 4*NUM$RCI;
                                                                                             01235000
               RCISSTORE (I-4) = RCISSTCRE (I);
                                                                                             01236000
                                                                                             01237000
           DC I = 4*J*1 TO 4*NUH$NCI;
                                                                                             01238000
               NCISSTORE (1-4) = NCISSTORE (1);
                                                                                             01239000
                                                                                             01240000
           DC I = 4*J+1 TO 4*NUM*GCI;
                                                                                             01241000
                                                                                             01242000
               GC1$STORE(I-4) = GC1$STCRE(I);
                                                                                             01243000
           END:
       ENC;
                                                                                             01244000
                                                                                             01245000
       DO CASE N;
                                                                                             01246000
                                                                                             01247000
           NUMSPCI = NUMSPCI - 1;
           NUMSRCI - NUMSRCI - 1;
                                                                                             01248000
           NUMSHCI - NUMSHCI - 1;
                                                                                             01249000
           NUPSGCI = NUMSGCI - 1;
                                                                                             01250000
       INC:
                                                                                             01251000
                                                                                             01252000
       LO CASE N;
                                                                                             01253000
```

A STATE OF THE PARTY OF

```
01254000
          PCISPOINTER * PCISPOINTER - 4;
RCISPOINTER * RCISPOINTER - 4;
          NCISPOINTER + NCISPOINTER - 4;
                                                                                      01256000
          GCISPOINTER - GCISPOINTER - 4;
                                                                                      01257000
                                                                                      01258000
                                                                                      01259000
       GO TC CONSUPLOAD1;
                                                                                      01260000
100/
                                                                                      01261000
/**/
                                                                                      01262000
/*
      MPU CI NUMBER TEST */
                                                                                      01263000
/**/
                                                                                      01264000
                                                                                      01265000
MPUCINUMTESTA: IF NUMSHCI + NUMSPCI + NUMSRCI>= 80
                                                                                      01266000
THEN GO TO CVERFLOWS;
                                                                                      01267000
  ELSE GG TO CONSUPLOADI;
                                                                                      01266000
/**/
                                                                                      01269000
144/
                                                                                      01270000
14
              OVERFLOW MESSAGE
                                                                                      01271000
/**/
                                                                                      01272000
                                                                                      01273000
/**/
OVERFLOW1: C$NODE = '0';
OVERFLOW2: EO I = 1 TO 27;
                                                                                      01274000
                                                                                      01275000
  GUTSTABLE ( TC + I ) = OVERFLOWHER (I - 1 );
                                                                                      01276006
                                                                                      01277000
END;
WAITING = WAITING + 1;
                                                                                      01278000
TC = TC + 47;
OUTTABLE (TC +1) = 03H;
                                                                                      01279000
                                                                                      01260000
OUTTABLE (TC +2) = 03h;
                                                                                      01261000
TC = TC + 4:
                                                                                      01262000
MSGCOURT (WAITING) = 29;
                                                                                      01263000
                                                                                      01284000
CC1 = 0:
GO TO CPULLINI;
                                                                                      01285000
1 + +/
                                                                                      01286000
10 4/
                                                                                      01287000
/* CHECKIN HUDE SET #/
                                                                                      01288000
/* */
/* */
                                                                                      01269000
                                                                                      01290000
CHECKINGOUSET1: IF CC1 < 9 THEN GC TO ADRERROR1;

IF CHARSTORE1(5) = '0' THEN C$HODE = '0';

IF CHARSTORE1(9) = '1' THEN C$HODE = '?';

ELSE C$HODE = 'C';
                                                                                      01291000
                                                                                      01292000
                                                                                      01293000
                                                                                      01294000
IF AUMSPCI + AUMSRCI + NUMSRCI >= 66 THEN GO TO OVERFLOW1;
                                                                                      01295000
    ELSE GC TO COMSUPLOAD1;
                                                                                      01296000
                                                                                      01297000
1001
                                                                                      01298000
/* HESET OF CHECKIN TABLES
                                                                                      01299000
1001
                                                                                      01300000
1001
                                                                                      01361000
PCIRESET: NUMSPC1 = 0;
FCISPOINTEN = 0;
                                                                                      01302000
                                                                                      01303000
            GO TO COMSUPLOAD 1:
                                                                                      01304000
ACIRESET: BUNSRCI - D;
                                                                                      01305000
            ACISPOINTER = 0;
                                                                                      01306000
            GG TO CONSUPLOAD1;
                                                                                      01307000
ACIRESET: NUMSHCI = 0;
NC1$POINTER = 0;
                                                                                      01308000
                                                                                      01309000
GO TO COMSUPLOAD;
GCIRESET: NUMSGCI = 0;
GCISPOINTER = 0;
                                                                                      01310000
                                                                                      01311000
                                                                                      01312000
                                                                                      01313000
/++/
144/
                                                                                      01314000
100/
              CONSUP MSG RETURN TO TERMINAL OPERATOR
                                                                                      01315000
                                                                                      01316000
```

```
01317000
COMSUPLOAD1: DO I = 1 TO CC1;
OUTSTABLE ( TC + I ) = CHARSTORE1(I);
                                                                                            01318000
                                                                                            01319000
END;
TC = TC + CC1;
WAITING = WAITING + 1;
                                                                                            01320000
                                                                                            01321000
                                                                                            01322000
OUTTABLE (TC +1) = 03H;
OUTTABLE (TC +2) = 03H;
                                                                                            01323000
                                                                                            01324000
TC = TC +2:
                                                                                            01325000
MSG$COUNT(WAITING) = CC1 +2;
                                                                                            01326000
NOCOMSUPLOAL1: CC1 = 0;
                                                                                            01327000
GO TO CPOLL 101;
                                                                                            01326000
/* */
/*
                                                                                            01329000
         EHRCH MESSAGE DUMP #/
                                                                                            01330000
100/
                                                                                            01331000
/**/
                                                                                            01332000
MODELHHOM1: DO I = 1 TO 30;
                                                                                            01333000
   CUTSTABLE ( TC + I ) = ERRORHDk(I - 1) ;
                                                                                            01334000
END;
TC = TC + 30;
                                                                                            01335000
                                                                                            01336000
HAITING . HAITING + 1;
                                                                                            01337000
RSGCOUNT (WAITING) = C;
HSGCOUNT (WAITING) = HSGCOUNT (WAITING) + 30;
                                                                                            01338000
                                                                                            01339000
cc1 = 6;
                                                                                            01340000
GO TO CPULL 101:
                                                                                            01341000
144/
                                                                                            01342000
/* */
/* TABLE THANSFERS */
                                                                                            01343000
                                                                                            01344000
/* */
/* MAKE UF HEADER */
                                                                                            01345000
                                                                                            01346000
/* */
TTR1: OUTTABLE1(TC1 + 1) = '*';
                                                                                            01347000
                                                                                            01348000
CUTTABLE 1(TC1 + 2) = CHARSTORE 1(9);
                                                                                            01349000
TC 11 A 17 I AL = 1C1;
DO I = 1 TO 3;
                                                                                            01350000
                                                                                            01351000
   OUTTABLE 1 ( TC1 + 2 + I) = CHARSTCRE1 ( I+9 ) ;
                                                                                            01352000
END;
/* */
/* SELECT TABLE */
                                                                                            01353000
                                                                                            01354000
                                                                                            01355000
                                                                                            01356000
IF CHARSTORE1(9) = 'X' THEN GO TO FITRI;
IF CHARSTORE1(9) = 'Y' THEN GO TO RITRI;
IF CHARSTORE1(9) = 'Z' THEN GO TO PITRI;
IF CHARSTORE1(9) = 'T' THEN GO TO PICLITRI;
                                                                                            01357000
                                                                                            01358006
                                                                                            01359000
                                                                                            01360000
  ELSE GO TO NODEERRCR1;
                                                                                            01361000
1 * 1
                                                                                            01362000
/* */
/* PCI TABLE LOAL */
                                                                                            01363000
                                                                                            01364000
1 +1
                                                                                            01365000
10 01
                                                                                            01366000
FTIR1: 15 NUMSPCI = 6 THEN GO TO TFRERRORS; WAITING1 = WAITING1 + 1;
                                                                                            01367000
                                                                                            01368000
ASGCOUNT 1 ( #AITING 1) = 0;
                                                                                            01369000
TC1 = TC1 + 5;
h = 4*hUMSFCI;
                                                                                            01370000
                                                                                            01371000
  DG I = 1 TO N;
OUT1APLE1( TC1 + I) = PCI$STOKE(I);
                                                                                            01372000
                                                                                            01373000
END;
TC1 = TC1 + N;
                                                                                            01374000
                                                                                            01375000
OUTTABLE 1 (TC 1+1) = G3H;
                                                                                            01376000
CUTTABLE 1 (IC1+2) = 03H;
                                                                                            01377000
TC1 = TC1 + 2;
                                                                                            01378000
MSGCOUNT 1 (WAITING 1) = TC1 - TC1 | NITIAL;
                                                                                            01379000
```

A selection in the selection of

2 62 LUNG

```
GO TO COMSUPLOAD1;
                                                                                  01380000
                                                                                  01381000
                                                                                  01382000
/* KCI TABLE LOAD */
                                                                                  01383000
                                                                                  01384000
1 * */
                                                                                  01385000
RTTR1: IF NUMSRCI = 0 THEN GO TO TFRERROR1;
                                                                                  01386000
WAITING1= WAITING1 + 1;
                                                                                  01387000
MSGCOUNT1(WAITING1) = 0;
                                                                                  01388000
TC1 = TC1 + 5;
                                                                                  01389000
A = 4*AUMSRCI;
                                                                                  01390000
  DO I = 1 TO A;
                                                                                  01391000
   CUTTABLE1( TC1 + 1) = RCI$STORE(I);
                                                                                  01392000
END;
TC1 = TC1 + N;
                                                                                  01393000
                                                                                  01394000
OUTTABLE 1 (1C 1+1) = 03H;
                                                                                  01395000
OUTTABLE 1 (1C1+2) = 03H;
                                                                                  01396000
101 = 101 + 2;
                                                                                  01397000
HSGCOURT ( #AITING 1) = TC1 - TC1 INITIAL;
                                                                                  01398000
                                                                                  01399000
GO TO COMSUFLOAD 1;
1 ./
                                                                                  01400000
/* */
/* NCI TABLE LGAD */
                                                                                  01401000
                                                                                  01402000
1 * */
                                                                                  01403000
10 01
NTINY: IF NUMERCI = C THEN GO TO THERRORY;
                                                                                  01405000
WAITING 1 - WAITING 1 + 1;
                                                                                  01406000
MSGCLUNT1(MAITING1) = 0;
TC1 = TC1 + 5;
                                                                                  01407000
                                                                                  01408000
h = 4 PhurshCI;
                                                                                  01409000
 DO I = 1 TO N;
OUTTABLE1( TC1 + I) = NCI$STORE(I);
                                                                                  01410000
                                                                                  01411000
END;
TC1 = TC1 + N;
                                                                                  01412000
                                                                                  01413000
OUITABLE 1 (IC1+1) = OJH;
                                                                                  01414000
OUTTABLE 1 (TC 1+2) = 03H;
                                                                                  01415000
TC1 = TC1 + 4:
                                                                                  01416000
HSGCOURT (WAITING 1) = TC1 - TC1INITIAL;
                                                                                  01417000
GO TO CONSULLOADI:
                                                                                  01418000
1 * 1
                                                                                  01419000
10 01
                                                                                  01420000
/* FOLL TABLE LOAD ( ALL CALL ) +/
                                                                                  01421000
10 01
                                                                                  01422000
10 01
                                                                                  01423000
POLLITAT: IF FT 1COUNT 1 <= 0 THEN GC TO TFRERHOFT; WAITING 1 = WAITING 1 + 1; MSGCOUNT 1 (-AITING 1) = C;
                                                                                  01424000
                                                                                  01425000
                                                                                  01426000
TC1 = TC1 + 5;

LG 1 = 1 10 PT 1COUNT1;

OUTTABLE1(TC1 + 1) = POLITABLE1(1);
                                                                                  01427000
                                                                                  01429000
END;
TC1 = TC1 + PI1COUNT1;
                                                                                  01430000
                                                                                  01431000
GUTTABLE 1 (TC 1+1) = 03H;
                                                                                  01432000
OUTTABLE 1 (101+2) = 03H;
                                                                                  01433000
TC1 = TC1 + 2;
                                                                                  11434000
MSGCOUNT 1 (MAITING 1) . TC1 - TC11N1TIAL;
                                                                                  01435000
GO TO COMSULLCADI:
                                                                                  01436000
100/
                                                                                  01437000
1001
                                                                                  01438000
/*
     TABLE THANSFER ERROR MESSAGE CUMP */
                                                                                  01439000
100/
                                                                                  01440000
TFHERROR1: LU : = 1 TO 33;
                                                                                  01441000
              CUITABLE (TC + I) = TFhERRORHDR (I-1);
                                                                                  01442000
```

```
END;
TC = TC + 33;
                                                                                          01443000
                                                                                          01444000
              BAITING - WAITING + 1:
                                                                                           01445000
             HSGCOUNT (WAITING) = 33;
                                                                                           01446000
                                                                                           01447000
            cc1 - 0;
              GO TO OPOLL 101;
                                                                                           01448000
                                                                                           01449000
                                                                                           01450000
/* TABLE CUMPS */
/* */
                                                                                           01451000
                                                                                           01452000
                                                                                           01453000
10 01
/* MAKE UF HEADER */
                                                                                           01454000
1: 1/
                                                                                           01455000
                                                                                           01456000
TBLDUMF1: IF CC1 < 12 THEN GO TO ADRERROR1;
OUTSTABLE(TC + 1) = 01H;
OUTSTABLE(TC + 2) = '3';
                                                                                           01457000
                                                                                           01458000
                                                                                           01459000
TC = TC + 2;
DO I = 1 Tc 3;
                                                                                           01460000
                                                                                           01461000
  QUISTABLE ( TC + I ) = CHARSTORE ( I+9 ) ;
                                                                                           01462000
LNL;
1C = 1C + 3;
                                                                                           01463000
                                                                                           01464000
WAITING = WAITING + 1;
                                                                                           01465000
MSGCOUNT(WAITING) = 5;
                                                                                           01466000
                                                                                           01467000
/* */
1 * */
                                                                                           01468000
                                                                                           01469000
/* SELECT TABLE */
/* */
/* */
                                                                                           01470000
                                                                                           01471000
IF CHARSTONE (9) = '0' THEN GO TO GTABLEDUMP1;
IF CHARSTONE (9) = 'P' THEN GO TO FTABLEDUMP1;
IF CHARSTONE (9) = 'K' THEN GO TO HTABLEDUMP1;
                                                                                           01472000
                                                                                           01473000
                                                                                           01474000
IF CHARSTORE (9) = 'C' THEN GO TO NTABLEDUMP 1;
IF CHARSTORE (9) = '2' THEN GO TO FCLLTABLEDUMP 1;
                                                                                           01475000
                                                                                           01476000
  ELSE GC TC MODEENROR1;
                                                                                           01477000
10 01
                                                                                           01478000
10 41
                                                                                           01479000
/* FRIGRITY */
                                                                                           01480000
10 01
                                                                                           01441000
                                                                                           01482000
PTABLEDUMP 1: LO I = 1 TO 29;
                                                                                           01483000
 OUTSTABLE (TC + I) = PIDHEADER (I - 1);
                                                                                           01484000
                                                                                           01485000
END;
TC = TC + 49;
                                                                                           01486000
HSGCOURT (WAITING) * MSGCOURT (WAITING) + 29;
                                                                                           01487000
IF NUMSPCI = 0 THEN GO TO CONSUPLOALT;
                                                                                           01486000
A = 1;

/* OUTPUTTING OF PCI TABLE- 8 CHECKINS PER LINE */

PTABLEBUILE1: DO I = 1 TO 8;
                                                                                           01490000
                                                                                           01491000
        DO J = 1 TO 4;
                                                                                           01492000
TC = TC + 1;
                                                                                           01493000
ESGCGUAT (MAITING) - MSGCOUNT (WAITING) + );
                                                                                           01494000
           CUISTABLE (TC) = PCISSTCHE (N);
                                                                                           01495000
            If a >= PCISPOINTER THEN GO TO COMSUPLOADI;
                                                                                           01496000
                                                                                           01497000
h = h + 1;
         ENL;
                                                                                           01498000
TC = TC + 1;
OUTSTABLE (TC) = 26H;
HSGCOURT (WAITING) = PSGCOURT (WAITING) + 1;
                                                                                           01459000
                                                                                           01500000
                                                                                           01501000
                                                                                           01502000
END;
TC = TC + 1;
                                                                                           01503000
OUTSTABLE (TC ) = ODH;
TC = TC + 1;
                                                                                           01564000
                                                                                           01505000
```

```
OUTSTABLE (TC ) = OAM;
HSGCOURT (WAITING) = HSGCOURT (WAITING) + 2;
                                                                                             01506000
                                                                                             01507000
GO TO PTABLEBUILD1:
                                                                                             01508000
144/
                                                                                             01509000
1441
                                                                                             01510000
                                                                                             01511000
/*
         RCUZINE
/**/
                                                                                             01512000
                                                                                             01513000
ATABLELUMP 1: DO I = 1 TO 28;
OUTSTABLE (IC + I) = RTDHEADER(I-1);
                                                                                             01514000
                                                                                             01515000
                                                                                             01516000
10 - 10 + 26;
                                                                                             01517000
MSGCOUNT (WAITING) = MSGCOUNT (WAITING) + 28;
                                                                                             01518000
IF NUMSECT = 0 THEN GO TO COMSUPLOADT;
                                                                                             01519000
N = 1;

/* OUTPUTIING OF RCI TABLE- 8 CHECKINS PER LINE */

KTABLEFUILL1: DO I = 1 TO 8;
                                                                                             01520000
                                                                                             01521000
                                                                                             01522000
LO J = 1 TC 4;
TC = TC+ 1;
                                                                                             01523000
                                                                                             01524000
MSGCOUNT (WAITING) = MSGCOUNT (WAITING) + 1;
                                                                                             01525000
OUTSTABLE (IC) = RCISSTORE (N)
                                                                                             01526000
IF N >= RCISPOINTER THEN GO TO CONSUPLOAD1;
                                                                                             01527000
N = h + 1;
                                                                                             01528000
IND;
TC = TC + 1;
GUT$TABLE (TC ) = 20h;
                                                                                             01529000
                                                                                             01530000
                                                                                             01531000
MSGCOUNT (WAITING) = MSGCOUNT (WAITING) + 1;
                                                                                             01532000
TC = TC + 1;

QUI$1ABLE (TC ) = QDH;
                                                                                             01533000
                                                                                             01534000
                                                                                             01535000
                                                                                             01536000
GUTSTABLE (TC ) = OAH;
                                                                                             01537000
MSGCOUNT (WAITING) = MSGCOUNT (WAITING) + 2;
                                                                                             01538000
GO TO RTABLEBUILD1:
                                                                                             01539000
104/
                                                                                             01540000
100%
                                                                                             01541000
                TRAFFIC
/*
                                                                                             01542000
/**/
                                                                                             01543000
ATABLEDUMP1: CO I = 1 TO 31;
                                                                                             01544000
OUTSTABLE (TC + I ) = NIDHEADER (1-1);
                                                                                             01545000
END;
                                                                                             01546000
TC = TC + 31;
H5GCOUNT (WAITING) = H5GCOUNT (WAITING) + 31;
IF NUMBEC = 0 THEN GO TO COMSUPLICATE;
                                                                                             01547000
                                                                                             01548000
                                                                                             01549000
h = 1;

/* OUTPUTTING OF ACT TABLE - 8 CHECKINS PER LINE */

NTABLEBUILD1: DO I = 1 TO 6;

DO J = 1 TO 4;

1C = TC + 1;
                                                                                             01550000
                                                                                             01551000
                                                                                             01552000
                                                                                             01553000
                                                                                             01554000
MSGCOUNT (WAITING) = MSGCOUNT (WAITING) + 1;
OUTSTABLE (TC) = NCISSTORE (N);
IF N >= NCISPOINTER THEN GO TO COMSUPLOAD1;
                                                                                             01555000
                                                                                             01556000
                                                                                             01557000
N = N + 1;
                                                                                             01558000
END;
TC = TC + 1;
                                                                                             01559000
                                                                                             01560000
HSGCOUNT (WAITING) = PSGCOUNT (WAITING) + 1;
OUTSTABLE (IC) = 20H;
                                                                                             01561000
                                                                                             01562000
                                                                                             01563000
END;
TC = TC + 1;
                                                                                             01564000
GUTSTABLE (TC) = ODH;
TC = TC + 1;
                                                                                             01565000
                                                                                             01566000
CUTSTABLE (TC) = OAh:
                                                                                             01567000
MSGCOURT (MAITING) = MSGCOURT (MAITING) + 2;
                                                                                             01568000
```

```
GO TG ATABLEBUILD1;
                                                                                    01569000
/**/
                                                                                    01570000
100/
                                                                                    01571000
/*
                                                                                    01572000
            GROUP CHECKIN
/++/
                                                                                    01573000
                                                                                    01574000
GTABLELUMP 1: DO I = 1 TO 26;
 OUTTABLE (IC + I) = GTDHEADER (I-1);
                                                                                    01576000
TC = TC + 26;
                                                                                    01577000
MSGCOUNT (MAITING) = MSGCOUNT (WAITING) + 26;
                                                                                    01578000
IF NUMSGCI = 0 THEN GO TO COMSUPLOADS;
                                                                                    01579000
h=1;
                                                                                    01580000
/* OUTPUTTING OF GCI TABLE - 8 CHECKINS PER LINE */
                                                                                    01581000
GTABLEBUILD1: DO I = 1 TO 8;

CO J = 1 TO 4;
                                                                                    01582000
                                                                                    01583000
              TC = TC + 1;
                                                                                    01584000
              MSGCOUNT (WAITING) = MSGCOUNT (WAITING) + 1;
                                                                                    01585000
              OUTTABLE (TC) = GCISSTORE(N);
IF N >= GCISPOINTER THEN GO TO COMSUPLOAD1;
                                                                                    01586000
          END;
TC = TC + 1;
                                                                                    01589000
                                                                                    01590000
          OLITABLE (TC) = 20H;
MSGCOUNT (WAITING) = MSGCOUNT (WAITING) + 1;
                                                                                    01591000
                                                                                    01592000
        END;
TC = TC + 1;
                                                                                    01593000
                                                                                    01594000
        OUTTABLE (TC) = ODH;
                                                                                    01595000
        TC = TC + 1;
                                                                                    01596000
        OUTTABLE (TC) = OAH;
                                                                                    01597000
        MSGCCUNT (HAITING) = MSGCOUNT (HAITING) + 2;
        GO 10 GTABLEBUILD1:
/**/
                                                                                    01600000
144/
                                                                                    01601000
/**/
                 POLL
                                                                                    01602000
                                                                                    01603000
/**/
                                                                                    01604000
POLLTABLEDUNP2: WAITING = WAITING + 1;
                                                                                    01605000
                  TOTALSHUHSCI = NUHSECI + NUMSECI + NUMSECI;
PTICOUNTI = 4 + TOTALSHUMSCI;
                                                                                    01606000
                                                                                     01607000
         RESET OF FOLLING HODE PARAMETERS
         POILSFLAG = 0:
         REFULLSFLAGSCOURT = 0;
         FOILSHSGSCOUNT = 0;
TDRSPOLLSTESSLOT = TDMSPOLLSTESSLOTSINITIAL;
                                                                                     01611000
                                                                                     01612000
POLITABLEDURP1: IF TOTALSRUMSCI = 0 THEN GO TO COMSUPLOAD1;
IF CHARSTORE1(13) = 'R' THEN POLISTABLESTXSFLAG = 01H;
                                                                                     01613000
                                                                                     01614000
    ELSE POLLSTABLESTESFLAG = OH;
                                                                                    01615000
DO 1 = 1 TC 26;
OUTSTABLE (TC + I ) = POLLDUMPHOR (I-1);
                                                                                    01616000
                                                                                     01617000
                                                                                     01618000
END;
TCINITIAL - TC;
TC = TC + 26;

/* OUTFUTTING OF FOLL TABLE- 8 CHECKINS PER LINE */
                                                                                     01620000
                                                                                     01621000
POLLDUMPBUILD1: LINECOUNT = 0;
                                                                                     01622000
N= 0;
                                                                                     01623000
      SLOT ASSIGNMENT HEADER - FIRST LINE
                                                                                     01624000
/*
144/
                                                                                     01625000
                                                                                     01626000
OUTTABLE (TC + 1) = 26H;
                                                                                     01627000
OUTTABLE (TC + 2) * FOLLSSLOT 1H;
                                                                                     01628000
OUTTABLE (TC + 3) = POLL$SLOT 11;
OUTTABLE (TC + 4) = 29H;
                                                                                     01629000
                                                                                     01630000
OUTTABLE (TC + 5) = 20H;
                                                                                     01631000
```

A SECTION AND ASSESSED.

```
TC = TC + 5;
                                                                                         01632000
LINECOUNT - LINECOUNT + 5;
                                                                                         01633000
LOADADA: DO I = 1 TO 4;
                                                                                         01634000
              OUTTABLE (TC+I) = POLITABLE 1 (N+I);
                                                                                         01635000
                                                                                         01636000
           END:
h = h+4;
                                                                                         01637000
OUTTABLE (TC+5) = 20H;
                                                                                         01638000
LINECOUNT = LINECOUNT + 5;
TC = TC + 5;
                                                                                         01639000
                                                                                         01640000
IF h >= FT1COUNT1 THEN GO TO ENDLOAD;
IF LINECOUNT < 45 THEN GO TO LOADADE;
                                                                                         01641000
                                                                                         01642000
   ELSE GO TO ENDICAT:
                                                                                         01643000
ENDLOAD: OUTTABLE (TC+1) = ODH;
                                                                                         01644000
           OUTTABLE (TC+2) = OAH;
                                                                                         01645000
           TC = TC + 2;
                                                                                         01646000
           LINECOUNT = 0;
                                                                                         01647000
           HSGCOUNT(NAITING) = HSGCOUNT(WAITING) + (TC-TCINITIAL);
                                                                                         01648000
           TCINITIAL = TC;
                                                                                         01649000
IF N >= PT 1COUNT 1 THEN GO TO POLISTABLESTX;
                                                                                         01650000
     SLOT ASSIGNMENT HEADER - AED'L LINES
GUTTABLE (TC + 1) = 28H;
                                                                                         01652000
LO CASE J:
                                                                                         01653000
   OUTTABLE (TC + 2) = POLL$SLOT1H;
                                                                                         01654000
   OUTTABLE (TC + 2) = POLL$SLOT2H;

GUTTABLE (TC + 2) = POLL$SLOT3H;

OUTTABLE (TC + 2) = POLL$SLOT4H;

OUTTABLE (TC + 2) = POLL$SLOT5H;

OUTTABLE (TC + 2) = POLL$SLOT6H;
                                                                                         01655000
                                                                                         01656000
                                                                                         01657000
                                                                                         01658000
                                                                                         01659000
END;
                                                                                         01660000
LO CASE J;
                                                                                         01661000
   GUTTABLE (TC + 3) = POLL$SLOT1L;
OUTTABLE (TC + 3) = POLL$SLOT2L;
                                                                                         01662000
                                                                                         01663000
   OUTTABLE (TC + 3) = POLL$SLOT3L;
OUTIABLE (TC + 3) = POLL$SLOT4L;
                                                                                         01664000
                                                                                         01665000
   GUTTABLE (TC + 3) = POLL$SLOT51;
                                                                                         01666000
   OUTTABLE (TC + 3) = POLL$SLOT61;
                                                                                         01667000
                                                                                         01668000
GUTTABLE (TC + 4) = 29H;
                                                                                         01669000
OUTTABLE (TC + 5) = 20H;
                                                                                         01670000
TC = TC + 5;
                                                                                         01671000
LINECOUNT = LINECOUNT + 5;
                                                                                         01672000
J = J + 1;
IF J > 5 THEN J = 0;
                                                                                         01673000
                                                                                         01674000
GO TO LOADALR;
                                                                                         01675000
144/
                                                                                         01676000
/**/
                                                                                         01677000
POLLSTABLESTX: IF FOLLSTABLESTXSFLAG <> 01H
                                                                                         01678000
                    THEN GO TO COMSUPLCADI;
                                                                                         01679000
         WAITING1 = WAITING1 + 1;
                                                                                         01680000
        DO 11 - 1 TO ESCOUNT (WAITING);
                                                                                         01681000
         OUTTABLE (TC 1+11) = OUTTABLE (TC+11-MSGCOUNT (WAITING));
                                                                                         01682000
                                                                                         01683000
                                                                                         01684000
          ALL-CALL HEADER SUBSTITUTION
                                                                                         01685000
         OUTIABLE (TC1 + 1) = 'S';
GUTIABLE (TC1 + 2) = 'L';
                                                                                         01686000
                                                                                         01687000
         OUTTABLE 1 (TC1 + 3) = '0';
GUTTABLE 1 (TC1 + 4) = 'T';
                                                                                         01688000
                                                                                         01689000
          CUITABLE 1 (TC1 + 5) = 2FH;
                                                                                         01690000
                                                                                         01691000
        msgcount (waiting ) = msgcount (waiting);
                                                                                         01692000
         TC1 - TC1 + MSGCOUNT (WAITING);
                                                                                         01693000
         POLISTABLESTESTASFLAG = 0;
                                                                                         01694000
```

```
GC 16 CONSUPLOAD1;
                                                                             01695000
144/
                                                                             01696000
/**/
                                                                             01697000
/*
       FGL1 TABLE SETUP
                                                                             01698000
/**/
                                                                             01699000
1001
                                                                             01700000
100/
                                                                             01701000
/*
       GET I C I DATA
                                                                             01702000
                                                                             01703000
144/
/**/
                                                                             01704000
FOLLTABLEBUILD1: IF AUMSPCI = 0 THEN GO TO RPOLIBUILD1:
                                                                             01705000
h = 4+hUESFCI;
                                                                             01706000
DO I = 1 TO N;
                                                                             01707000
 POLITABLE 1(I) . PCISSTORE (I);
                                                                             01708000
END;
                                                                             01709000
100/
                                                                             01710000
/**/
                                                                             01711000
       GET R C I DATA
/*
                                                                             01712000
100/
                                                                             01713000
144/
                                                                             01714000
APOLIBUILD 1: IF NUMSACI = 0 THEN GO TO APOLIBUILD 1;
                                                                             01715000
h = 4+hUHShCI;
FULLFOINTERT = 4 * NUMSPCI;
DO I = 1 TC N;
                                                                             01718000
 FULLTABLE 1 (POLLPOINTER 1 + I) = RCISSTURE (1);
                                                                             01719000
END:
                                                                             01720000
144/
                                                                             01721000
/**/
                                                                             01722000
/* GET N C I DATA
                                                                             01723000
/**/
                                                                             01724000
/**/
                                                                             01725000
APOLLBUILD 1: IF NUMSACI . O THEN GO TO GPOLLBUILD 1;
                                                                             01726000
h = 4*hUMShCI;
                                                                             01727000
FOLLPOINTER1 = 4 * (NUMSPCI + NUMSECI);
                                                                             01728000
DO I = 1 TC N;
                                                                             01729000
 POLITABLE 1 ( POLIPOINTER 1 + I ) = NCISTORE (1);
                                                                             01730000
END:
                                                                             01731000
144/
                                                                             01732000
144/
                                                                             01733000
/*
        GET GCI DATA
                            */
                                                                             01734000
                                                                             01735000
/**/
                                                                             01736000
GPOLLBUILD 1: IF NUMSGCI = 0 THEN GC TO POLLTABLEDUMP 2;
                                                                             01737000
      h = 4+NUMSGCI;
                                                                             01738000
      POLLFOINTER1 = 4 * (NUMSPC1 + NUMSRCI + NUMSRCI);
                                                                             01739000
      DO I = 1 TO N;
                                                                             01740000
        FOILTABLE 1 (POLLPOINTER 1 + 1) = GCISSTORE (I);
                                                                             01741000
      EDC:
                                                                             01742000
      GO TC POLLTABLEDUMP2:
                                                                             01743000
100/
                                                                             01744000
/**/
/*
                                                                             01745000
       OUTFUT
                                                                             01746000
/**/
                                                                             01747000
OFOLL 101: 11 DELAY1 = 1 THEN GO TO CELAYSROUTINE1;
                                                                             01748000
           IF WAITING 1 <= 0 THEN GO 10 IPOLL1;
                                                                             01749000
           IF TC1 > 0 THEN GO TO EOPTEST1;
ELSE GO TO IPOLL1;
                                                                             01750000
                                                                             01751000
EOMIESTI: IF OUTCOUNT! < MSGCOUNT!(1) THEN GO TO SPACETEST!;
                                                                             01752000
            ELSE CELAY1 = 1:
                                                                             01753000
SPACETEST1: SPACE1 = INPUT 105$DATA NOR OFFH;
                                                                             01754000
SPACE1 = SPACE1 AND G2H;
IF SPACE1 = 02H THEN GO TO OUTCHAR1;
                                                                             01755000
                                                                             01756000
   ELSE DELAY! . 0;
                                                                             01757000
```

A Marian Bank Bank

```
GO TO IFOIL1;
                                                                                  01758000
/**/
                                                                                  01759000
144/
                                                                                  01760000
/*
          DELAY ROUTINES
                                                                                  01761000
144/
                                                                                  01762000
/**/
                                                                                   01763000
DELAYSHOUTINE1: CLK1 = IMPUT(230);
                                                                                   01764000
       CLA1 = CLK1 AND OTH;
                                                                                   01765000
       IF CIA1 = CIRP1 THEN GO TO IPCLL1;
                                                                                  01766000
       CLECHT1 = CLECHT1 + 1;
                                                                                   01767000
       CLKP1 = CLK1;
                                                                                   01768000
                                                                                   01769000
       IF CIKCATI < BUFFERSDELAY1 THEN GO TO IPOLL1;
       CUIPUT (235) = 08H;
IF TEMSFLAG <> '1' THEN GO TO NOWSTDMSDELAY1;
                                                                                   01770000
                                                                                  01771000
TORSCELAYS &CUTINE 1: CLK3 = IMPUT (236);
                                                                                   01772000
      CLK3 = CLK3 AND O1H;
1F CLK3 = CLKP3 THEN GO TO 1FCLL1;
                                                                                   01773000
                                                                                   01774000
       CLKCNI3 = CLKCNI3 + 1;
                                                                                   01775000
       CLKP3 = CLK3;
                                                                                   01776000
       IF CLECKT3 < SLOTSGUARDSTIME THEN GO TO IPOLL1;
                                                                                   01777000
                                                                                   01778000
       DELAY1 = 0:
       LLACA11 = 0;
                                                                                   01779000
       CLKCN12 = 0;
                                                                                   01780000
       CLKC N13 = 0;
                                                                                   01761000
       GO TO IPOLL1;
                                                                                   01782000
NUMSTDESCELAY1: CLK2 = INPUT (230);
C1K2 = CLK2 AND OTH;
                                                                                   01783000
                                                                                   01784000
          IF CLK2 * CLKP2 THEN GO TO IPOLL1;
CLKChT2 * CLKChT2 + 1;
                                                                                   01785000
                                                                                   01786000
          CINF2 = CIN2;
                                                                                   01787000
          IF CLKCHT2 < NONSTDMSMSGSLELAY1 THEN GO TO IFOLL1;
                                                                                   01788000
          IF CC = 0 THEN DELAY1 = CH;
                                                                                   01789000
                                                                                   01790000
            ELSF DELAY1 = 01H;
          IF NONSTOMSESGEDELAY? <= 3CH THEN GO TO HX$BUSY$OVERRIDE?;
                                                                                   01791000
          CINCRII = 0;
                                                                                   01792000
          CLKCNT2 = 0;
                                                                                   01793000
          CLECKI3 = 0;
                                                                                   01794000
          GU TO IFOLL1;
                                                                                   01795000
1441
                                                                                   01796000
                                                                                   01797000
AXIBUSTICVERRIDET: DELAYT = 0;
                                                                                   01798000
          DC CASE FOLLSFLAG;
                                                                                   01799000
              NONSTEMSHSGSDELAY1 = 0;
                                                                                   01600000
                                                                                   01801000
              NONSTONSMSGSDELAY1 = 36h;
                                                                                   01802000
          ENE:
          CIRCUIT = C;
                                                                                   01603000
          CLECHT2 = 0;
                                                                                   01804000
          CLECHT3 = C;
                                                                                   01605000
                                                                                   01806000
          GC TO IFOLL1:
                                                                                   01807000
144/
100/
                                                                                   01808000
OUTCHART: CUTPUT (235) = 09H;
OUTFUT 1011LATA = OUTTABLE 1 (OUTCCURT) XOR OFFH;
                                                                                   01809000
                                                                                   01810000
GUICGUNI 1 = OUTCOUNT 1 + 1;
SHIFTECHN1: IF OUTCOUNT 1 <= MSGCGUN1 1(1) THEN GO TO IPOLL 1;
ILSE DG I1 = OUTCOUNT 1 TO TC 1;
                                                                                   01811000
                                                                                   01812000
                                                                                   01613000
GUTTABLE1 ( I1 +1 - GUTCOUNT 1) = OUTTABLE1(I1);
                                                                                   01814000
                                                                                   01815000
ENU;
TO 11 = 1 TO WAITING1;
                                                                                   01816000
                                                                                   01617000
MSGCOUNT1(11) = MSGCOUNT1(11 + 1);
ENL;
TC1 = TC1 - ( OUTCOUNT1 - 1 ) ;
                                                                                   01618000
                                                                                   01819000
MSGCOUNT I (WAITING 1) = U;
                                                                                   01820000
```

```
WAITING1 = WAITING1 - 1;
OUTCOUNT1 = 1;
IF WAITING1 = 0 THEN TC1 = 0;
GO TO IFOLL1;
OUIT: OUTPUT(2) = CC;
CUIPUT(2) = CC1;
CUIPUT(2) = MSGCOUNT(1);
OUTPUT(2) = MSGCOUNT(1);
OUTPUT(2) = TC;
CUTPUT(2) = TC1;
CUTPUT(2) = TC1;
CUTPUT(2) = OUTCOUNT1;
CUTPUT(2) = OUTCOUNT1;
CUTPUT(2) = WAITING;
CUTPUT(2) = WAITING1;
OUTPUT(2) = WHSPC1;
OUTPUT(2) = NUMSPC1;
OUTPUT(2) = NUMSPC1;
CUTPUT(2) = NUMSPC1;
CUTPUT(2) = NUMSCC1;
CUTPUT(2) = NUMSCC1;
CUTPUT(2) = PTICOUNT1;
EOF
                                                                                                                                                                                                                                                                                01821000
01822000
01823000
                                                                                                                                                                                                                                                                                01824000
01825000
                                                                                                                                                                                                                                                                                01826000
01827000
                                                                                                                                                                                                                                                                                 01828000
                                                                                                                                                                                                                                                                                 01829000
                                                                                                                                                                                                                                                                                 01830000
                                                                                                                                                                                                                                                                                 01831000
                                                                                                                                                                                                                                                                                 01832000
                                                                                                                                                                                                                                                                                 01833000
                                                                                                                                                                                                                                                                                  01834000
                                                                                                                                                                                                                                                                                  01835000
                                                                                                                                                                                                                                                                                 01836000
                                                                                                                                                                                                                                                                                 01837000
                                                                                                                                                                                                                                                                                 01638000
                                                                                                                                                                                                                                                                                 01839000
  EOF
                                                                                                                                                                                                                                                                                 01840000
```

とう 一般の はない 本のではない 日本のののできる というない はいけい

GLOSSARY

AFSATCOM Air Force Satellite Communications

AFSC Air Force Systems Command

ASR automatic send receive

BCD binary coded decimal

b/s bits per second

CINC Commander-in-Chief

CMOS complementary metal-oxide semiconductor

COMSUP communications supervisory

CP command post

CPU central processing unit

CRT cathode ray tube

EAM emergency action message

EDAC error detection and correction

EPROM eraseable programmable read-only memory

ESD Electronic Systems Division

FLTSATCOM Fleet Satellite Communications

FSA free storage area

FSK frequency shift keying

IC integrated circuit

I/O input/output

ISA instruction storage area

LSI large scale integration

GLOSSARY (Continued)

MI message indicator

MPU message processor unit

NB narrowband

NCA National Command Authority

NCS net control station

OW orderwire

PROM programmable read-only memory

R/T receiver/transmitter

RX receive

SBC single board computer

SDS Satellite Data System

SIOP Single Integrated Operations Plan

SSDA synchronous serial data adapter

TDM time division multiplex

TTL transistor-transistor logic

TX transmit

UHF ultra high frequency

VSA variable storage area

WB wideband

WWABNCP Worldwide Airborne Command Post

